

DENEYAP KART PROJE ÖRNEKLERİ

1. Proje Adı: Deneyap Kart ile Google Gemini Yapay Zeka İletişimi

Gerekli Malzemeler:

- Deneyap Kart 1A v2
- Type-C kablosu
- Bilgisayar (Deneyap Kart IDE kurulu)
- Wi-Fi ağına ait kullanıcı adı ve şifre
- Google hesabı (Gemini API anahtarı almak için)

Proje Devre Şeması:

Bu projede herhangi bir ek devre kurulumuna gerek yoktur. Deneyap Kart 1A v2, Type-C kablosu ile doğrudan bilgisayara bağlanarak kullanılacaktır. Bu nedenle Fritzing çizimi gerekmemektedir.

Proje Kodları :

```
#include <WiFi.h>
#include <HTTPClient.h>
#include <ArduinoJson.h>

const char* ssid = "WIFI_ADINIZ";           // Wi-Fi adınızı buraya yazın
const char* password = "WIFI_SIFRENIZ";     // Wi-Fi şifrenizi buraya yazın
const char* Gemini_Token = "API_ANAHTARINIZ"; // Google API anahtarınızı
buraya yapıştırın

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  Serial.print("Wi-Fi'ye bağlanılıyor...");
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("Bağlandı!");
}

void loop() {
  if (WiFi.status() == WL_CONNECTED) {
    Serial.println("Sorunuzu yazın: ");
    while (Serial.available() == 0) {
      delay(100);
    }
  }
}
```

```
}  
String question = Serial.readStringUntil('\n');  
askGemini(question);  
}  
delay(1000);  
}  
  
void askGemini(String question) {  
  HTTPClient http;  
  String url =  
  "https://generativelanguage.googleapis.com/v1beta/models/gemini-1.5-flash-latest:generateContent?key=" + String(Gemini_Token);  
  http.begin(url);  
  http.addHeader("Content-Type", "application/json");  
  
  String payload = "{\"contents\": [{\"parts\": [{\"text\": \"" + question +  
  "\"}]]]}";  
  int httpResponseCode = http.POST(payload);  
  
  if (httpResponseCode > 0) {  
    String response = http.getString();  
    DynamicJsonDocument doc(2048);  
    deserializeJson(doc, response);  
    const char* answer = doc["candidates"][0]["content"]["parts"][0]["text"];  
    Serial.println("Cevap: ");  
    Serial.println(answer);  
  } else {  
    Serial.println("Hata oluştu: " + String(httpResponseCode));  
  }  
  http.end();  
}
```

Adım Adım Yapılacaklar:

1. Bilgisayarı Hazırlayın:

- Bilgisayarınızda bir internet tarayıcısı açın (örneğin, Google Chrome).
- <https://www.deneyapkart.org/> adresine gidin.
- “Deneyap Kart IDE” programını indirin ve kurun. Kurulum bittiğinde, programı açın; masaüstünde bir simge görünecek.

2. Google API Anahtarı Alın:

- Tarayıcınızda yeni bir sekme açın ve <https://aistudio.google.com/> adresine gidin.
- Gmail adresiniz ve şifrenizle Google hesabınıza giriş yapın.

- Sayfada “Get API Key” (API Anahtarı Al) yazan bir düğmeye tıklayın.
- Size verilen uzun bir kodu (harf ve rakamlardan oluşan) kopyalayın. Bu kodu bir not defterine ya da bir kâğıda kaydedin, çünkü sonra kullanacağız.

3. Wi-Fi Bilgilerinizi Not Edin:

- Evde veya okulda kullandığınız Wi-Fi’nin adını ve şifresini bir kâğıda yazın.
Örneğin:
 - Wi-Fi adı: EvWiFi
 - Şifre: 12345678
- Bu bilgileri kodda kullanacağız.

4. Kodu Düzenleyin:

- Yukarıdaki Proje Kodlarını kopyalayın.
- Deneyap Kart IDE’yi açın ve kodları programın boş alanına yapıştırın.
- Kodda şu üç yeri kendi bilgilerinizle değiştirin:
 - `const char* ssid = "WIFI_ADINIZ";` yazan yere Wi-Fi adınızı yazın (örneğin, `const char* ssid = "EvWiFi";`).
 - `const char* password = "WIFI_SIFRENIZ";` yazan yere Wi-Fi şifrenizi yazın (örneğin, `const char* password = "12345678";`).
 - `const char* Gemini_Token = "API_ANAHTARINIZ";` yazan yere kopyaladığınız Google API anahtarını yapıştırın (örneğin, `const char* Gemini_Token = "abc123xyz";`).

5. Gerekli Kütüphaneyi Yükleyin:

- Deneyap Kart IDE’de üst menüde “Araçlar” yazan yere tıklayın.
- “Kütüphane Yöneticisi”ni seçin.
- Açılan pencerede arama çubuğuna “ArduinoJson” yazın.
- “ArduinoJson” kütüphanesini bulup yanındaki “Yükle” düğmesine tıklayın.
Yükleme bitene kadar bekleyin.

6. Deneyap Kart’ı Bilgisayara Bağlayın:

- Deneyap Kart 1A v2’yi Type-C kablosuyla bilgisayara takın (telefon şarj kablosu gibi düşünün).
- IDE’de üst menüden kartınızı seçin: “Deneyap Kart 1A v2”yi bulup tıklayın.
- Aynı menüde “Port” kısmından kartınızın bağlı olduğu portu seçin (örneğin, COM3). Eğer emin değilseniz, kartı takıp çıkarın ve port listesinin değişip değişmediğine bakın.

7. Kodu Karta Yükleyin:

- IDE’de sağ üst köşede bir ok işareti (“Yükle” düğmesi) var, ona tıklayın.
- Kodun karta yüklenmesi birkaç saniye sürecek. Ekranda “Yükleme tamamlandı” yazısını görene kadar bekleyin.

8. Soru Sorun ve Cevap Alın:

- IDE’de üst menüden “Seri Monitör”ü açın (genellikle büyüteç simgesi).
- Seri Monitör açıldığında, altta bir kutucuk göreceksiniz. “Sorunuzu yazın:” yazısını gördüğünüzde, bir soru yazın (örneğin, “En büyük gezegen hangisidir?”) ve klavyede Enter tuşuna basın.

- Biraz bekleyin, yapay zekanın cevabı Seri Monitör’de görünecek.

9. Daha Fazla Soru Sorun:

- Farklı sorular sorarak yapay zekayı test edin. Mesela:
 - “Türkiye’nin en uzun nehri hangisidir?”
 - “En hızlı araba nedir?”
- Her sorudan sonra cevabı okuyun ve yeni bir soru sorabilirsiniz.

Hatırlatma:

- Google API anahtarını doğru kopyalayın ve kodda belirtilen yere tırnak içinde yapıştırın.
- ArduinoJson kütüphanesinin Deneyap Kart IDE üzerinden yüklendiğinden emin olun. Bu kütüphane, Gemini’den gelen cevapları işlemek için gereklidir.
- Deneyap Kart’ın Type-C kablosuyla bilgisayara düzgün bağlandığından ve doğru portun IDE’de seçildiğinden emin olun.
- Seri Monitör’ü açarken baud rate’in 115200 olarak ayarlandığını kontrol edin.
- İnternet bağlantısının stabil olduğundan emin olun; bağlantı kesintileri projenin çalışmasını engelleyebilir.
- Wi-Fi adı ve şifresinin doğru girildiğinden emin olun. Yanlış bilgiler bağlantı sorunlarına yol açabilir.

Ek Dosya Linkleri:

Google Gemini API anahtarı alma: <https://aistudio.google.com/>

DENEYAP KART

2. Proje Adı: Akıncı Simülasyonu

Deneyap Kart ile Akıncı Simülasyonu projesinin amacı, uçuş kontrol sistemlerinin temel prensiplerini öğrenmek ve ivmeölçer (accelerometer) ile jiroskop (gyroscope) sensör verilerini kullanarak bir Akıncı simülasyonu oluşturabilmektir. Bunu da Deneyap Kartta dahili olarak bulunan IMU sensörü yardımıyla yapmaktadır.



Gerekli Malzemeler:

Bu proje için sadece Deneyap Kart ve bilgisayarla bağlantısı için bir kablo yeterlidir.

Yapım Aşaması:

1. Dosya Yükleme:

Proje için aşağıdaki linkte verdiğimiz dosyaların yüklenmesi.

2. Deneyap Kart Kodu:

Deneyap Kartı bilgisayarımıza bağlayarak ide_kod dosyasındaki kod Deneyap Karta yüklenmelidir. Bu işlemden sonra ide programı kapatılabilir.

3. Processing Programı:

plane_modelling kod dosyası processing programında çalıştırılmalıdır.

Proje Linki:

Projenin bütün detaylarına aşağıdaki linkten ulaşabilirsiniz:

https://github.com/edacmakkk/akinci_simulation/tree/main

Proje Kodları:

Deneyap karta yüklenmesi gereken kod linki:

https://github.com/edacmakkk/akinci_simulation/blob/main/ide_kod/ide_kod.ino

```
#include "lsm6dsm.h"
#define delaysms 100
LSM6DSM IMU;
// float elapsedTime, currentTime, previousTime;
// float yaw;

void setup() {
  Serial.begin(115200);
  IMU.begin();
}

void loop() {
  float accelX = IMU.readFloatAccelX() * 2;
  float accelY = IMU.readFloatAccelY() * 2;
  float GyroZ = IMU.readFloatGyroZ();
  Serial.print(accelX);
  Serial.print(",");
  Serial.print(accelY);
  Serial.print(",");
  Serial.println(GyroZ);
  delay(delaysms);
}
```

Processing programında çalıştırılması gereken kod linki:

https://github.com/edacakmakk/akinci_simulation/blob/main/plane_modelling.pde

```
import processing.serial.*;
PShape airplane; // Uçak modelini tanımlama
PImage cloudImg;

float backgroundX = 0; // Arka planın X eksenindeki konumu
float backgroundY = 0; // Arka planın Y eksenindeki konumu
float sensitivity = 2.0; // Arka plan hareketinin hassasiyeti

Serial myPort;
float x, y, z;

void setup() {
  size(1920, 1080, P3D); // 3D desteğini aktif hale getirme
```

```
String portName = Serial.list()[0]; //Kartın takılı olduğu portun kaçınıcı
sırada olduğunu gösterir. Birden fazla port kullanılıyorsa kartın hangi sırada
olduğuna aygıt yöneticisinden bakılmalıdır.
myPort = new Serial(this, portName, 115200);
myPort.bufferUntil('\n');
// Uçak modelini yükleme
airplane = loadShape("akincii.obj");
//cloudImg = loadImage("bulut.jpg");
airplane.scale(40); // Modeli uygun boyutta ölçekleme
}

void draw() {
    background(200);
    pushMatrix();
    //translate(width/2, height/2); // Bulut resmini ekranın ortasına koy
    //imageMode(CENTER);
    //image(cloudImg, 0, 0, 800, 600); // Bulut resmini ekran boyutuna uygun
şekilde çiz
    popMatrix();
    lights(); // Işık ekleyerek 3D objelerin daha iyi görünmesini sağlama
    translate(width/2, height/2);
    rotateX(radians(y));
    rotateY(radians(z));
    rotateZ(radians(x));
    shape(airplane); // Uçak modelini ekrana çizme
}

void serialEvent(Serial myPort) {
    String inString = myPort.readStringUntil('\n');
    if (inString != null) {
        println(inString);
        String[] values = split(trim(inString), ',');
        if (values.length == 3) {
            x = float(values[0]) * 45;
            y = float(values[1]) * 45;
            z = float(values[2]) * -0.9;
        }
    }
}
```

3D Simülasyon dosyası:

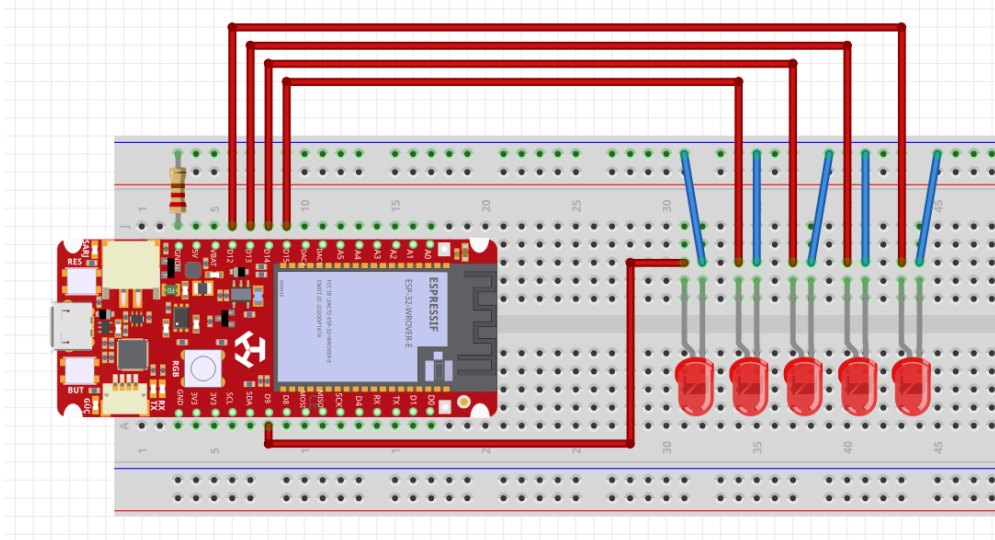
https://github.com/edacakmakk/akinci_simulation/blob/main/akincii.obj

3. Proje Adı: Deneyap Kart ve Kamera ile Parmak Kontrollü Ai LED Projesi

Gerekli Malzemeler:

- 1 Adet Deneyap Kart 1A
- 5 adet LED
- 1 Adet 220 ohm Direnç
- 10 Adet Jumper Kablo

Proje Devre Şeması:



Proje Kodları :

```
#include <WiFi.h>
#include <WiFiUdp.h>
#include <esp_wifi.h>

const char* ssid = "DeneyapKartAP";
const char* password = "12345678.";
const char* udpAddress = "0.0.0.0"; // UDP istemcisinin IP adresi
const int udpPort = 14537;
int fingers[5] = { 0, 0, 0, 0, 0 };
int lights[5] = { D9, D12, D13, D14, D15 }; // Deneyap Kart (Hangi kartı kullanılıyorsa, Karta uygun D pin bağlantıları gerçekleştirilmelidir.)
// int lights[] = {D13, D14, D12, D10, A7 }; // Deneyap Kart 1A v2
// int lights[] = {D2, D3, D4, D5, D6 }; // Deneyap Mini v2
WiFiUDP udp;

void setup() {
  Serial.begin(115200);
  Serial.println("Başladı");
```

```
Serial.begin(115200);
delay(2000);
// ESP32'yi Access Point olarak başlat
WiFi.softAP(ssid, password);
Serial.println("Access Point olusturuldu.");
Serial.print("IP Adresi: ");
Serial.println(WiFi.softAPIP());
Serial.println("WiFi'ye bağlanıldı");
Serial.println(WiFi.localIP());
// Pin çıkışları ayarlanıyor
for (int i = 0; i < 5; i++) {
    pinMode(lights[i], OUTPUT);
    digitalWrite(lights[i], LOW);
}
// UDP'yi başlat
udp.begin(udpPort);
xTaskCreate(receiveDataTask, "ReceiveUdpData", 8096, NULL, 1, NULL);
}

void loop() {
    delay(1000);
}

void receiveDataTask(void* parameter) {
    char incomingPacket[255]; // Gelen veriyi saklamak için buffer
    while (true) {
        // Gelen UDP paketlerinin olup olmadığını kontrol et
        int packetSize = udp.parsePacket();
        if (packetSize) {
            // Paket boyutunu ve gönderici bilgilerini al
            Serial.printf("Paket boyutu: %d\n", packetSize);
            int len = udp.read(incomingPacket, 255); // Gelen veriyi buffer'a oku
            if (len > 0) {
                incomingPacket[len] = 0; // String sonu null karakteri ekle
            }
            Serial.printf("Gelen UDP verisi: %s\n", incomingPacket); // Veriyi
            ekrana yazdır
            parseUDPData(incomingPacket, fingers);
            for (int i = 0; i < 5; i++) {
                Serial.print("Parmak ");
                Serial.print(i);
                Serial.print(": ");
                Serial.println(fingers[i]);
            }
        }
    }
}
```

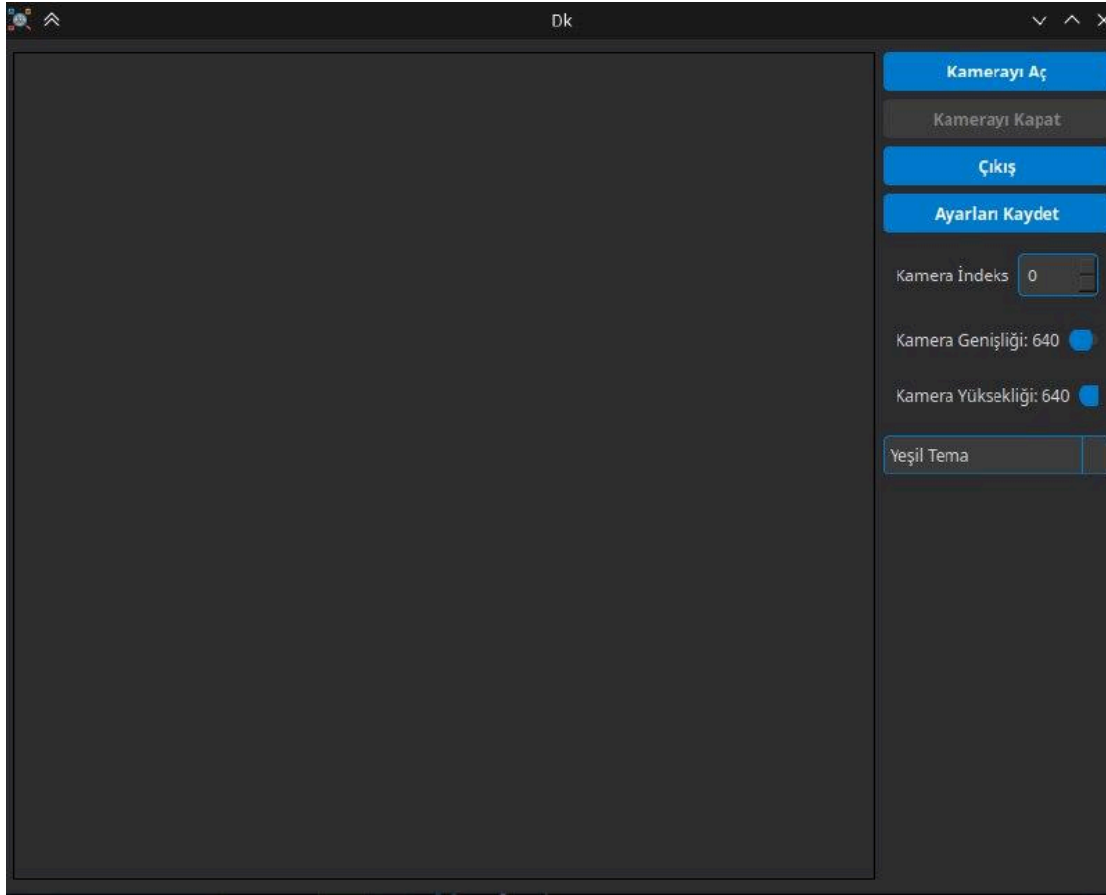
```
    }  
    finger_to_light();  
    vTaskDelay(10 / portTICK_PERIOD_MS); // 30 fps için 33ms bekle  
    }  
}  
}  
  
void parseUDPData(const char* udpData, int fingers[5]) {  
    // Köşeli parantezleri ve boşlukları temizle  
    char dataCopy[50];  
    strncpy(dataCopy, udpData, sizeof(dataCopy) - 1);  
    dataCopy[sizeof(dataCopy) - 1] = '\\0';  
    // Köşeli parantezleri kaldır  
    char* cleanData = dataCopy;  
    if (cleanData[0] == '[') {  
        cleanData++; // İlk karakteri atla  
    }  
    // Son parantezi kaldır  
    int len = strlen(cleanData);  
    if (len > 0 && cleanData[len-1] == ']') {  
        cleanData[len-1] = '\\0';  
    }  
    // Veriyi virgüllerle ayır  
    char* token = strtok(cleanData, ",");  
    int index = 0;  
    // Veriyi int'e çevirip fingers dizisine at  
    while (token != nullptr && index < 5) {  
        // Boşlukları atla  
        while (*token == ' ') {  
            token++;  
        }  
        fingers[index] = atoi(token); // String'i int'e dönüştür  
        token = strtok(nullptr, ","); // Bir sonraki kısmı al  
        index++;  
    }  
}  
  
void finger_to_light() {  
    for (int i = 0; i < 5; i++) {  
        if (fingers[i] == 1) {  
            digitalWrite(lights[i], HIGH);  
        } else {
```

```
digitalWrite(lights[i], LOW);  
}  
}  
}
```

Adım Adım Yapılacaklar:

- Bilgisayarınızda bir internet tarayıcısı açın (örneğin, Google Chrome).
- <https://www.deneyapkart.org/> adresine gidin.
- “Deneyap Kart IDE” programını indirin ve kurun. Kurulum tamamlandığında programı açın; masaüstünde bir simge oluşacak.
- Devre Şemasını Kurun:
5 adet LED’in uzun bacaklarını (anot) Deneyap Kart’taki D9, D12, D13, D14 ve D15 pinlerine bağlayın. LED’lerin kısa bacaklarını (katot) ise kartın GND pinine bağlayın.
- Yukarıda verilen proje için gerekli kodu Deneyap Kart IDE üzerinden kartınıza yükleyin.
- Kod kart yüklendikten sonra Wi-Fi ayarlarından “DeneyapKartAP” isimli WiFi ağına bağlanın. Wifi şifresi: **12345678**.
- https://mth.tc/DK_AI_Projesi adresindeki uygulamayı bilgisayarınıza indirip çalıştırın. (Bu uygulama exe uzantılı olduğu için yalnızca windows bilgisayarlarda kullanılabilir. Farklı bir işletim sistemine sahip bilgisayarlar için [Deneyap Kart Projeler](#) sayfasından gerekli proje adımları gerçekleştirilmelidir.)
- Yukarıdaki proje kodunu karta attıktan sonra indirmiş olduğunuz DK isimli uygulamayı çalıştırın. Uygulamanın arayüzü aşağıdaki gibidir. “Kamerayı Aç” kısmına tıkladığında görüntü işlenmeye başlanacaktır.

DENEYAP KART



Hatırlatma:

- LED'lerin doğru pinlere bağlandığından emin olun. Yanlış pin bağlantısı LED'lerin çalışmamasına neden olabilir.
- Wi-Fi adı ve şifresinin doğru olduğundan emin olun; aksi halde kart internete bağlanamaz.
- Seri Monitör'ü açarken baud rate'in 115200 olduğundan emin olun.

Ek Dosya Linkleri:

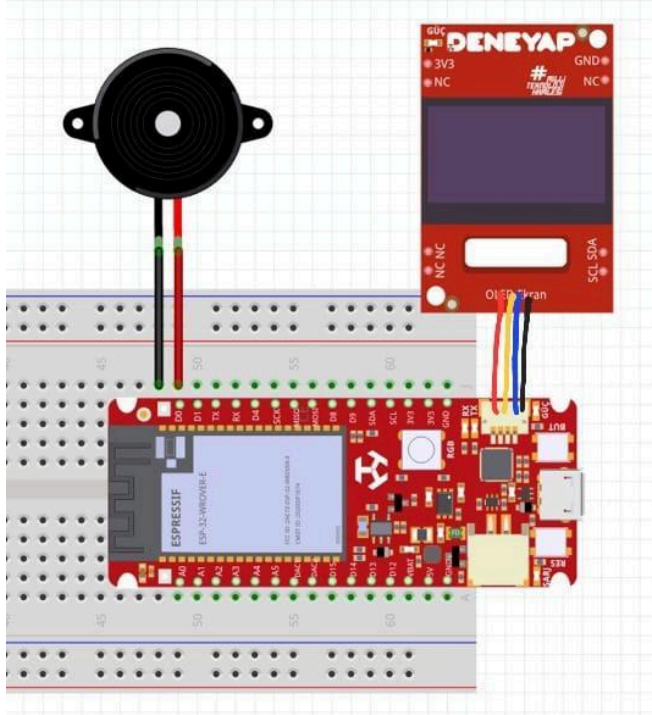
- Deneyap Kart IDE indirme bağlantısı: <https://www.deneyapkart.org/>
- Python mediapipe kütüphanesi: <https://pypi.org/project/mediapipe/>
- Projenin tüm kodları ve detayları:
https://github.com/halak0013/DK_Camera_Finger_to_Light

4. Proje Adı: Seri Monitor ile Sayaç Yapımı

Gerekli Malzemeler:

- 1 adet Deneyap Kart 1A
- 1 adet Deneyap Oled Ekran
- 1 adet Buzzer
- 1 adet I2C Bağlantı Kablosu

Proje Devre Şeması:



Proje Kodları :

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

// OLED ekran boyutları
#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64

// OLED ekran objesini tanımla
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, -1);
// Buzzer pin tanımla
```

```
#define BUZZER_PIN D12

int hours;
int minutes;
int seconds;
int totalSeconds;
String inputString = "";

void setup() {
    Serial.begin(115200);
    // Buzzer pinini çıkış olarak ayarla
    pinMode(BUZZER_PIN, OUTPUT);
    digitalWrite(BUZZER_PIN, LOW); // Buzzer başlangıçta kapalı
    // OLED ekranı başlat
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3D)) {
        Serial.println(F("OLED ekran bulunamadı"));
        while (true);
    }
    // Ekranı temizle
    display.clearDisplay();
    display.setTextColor(WHITE);
    // Süre ayarlama mesajını göster
    display.setTextSize(1); // Text boyutu
    // Mesajları ayarla
    String line1 = " SURE";
    String line2 = "GIRISI";
    // Her bir satır için merkezdeki X ve Y koordinatları
    int16_t textX = (SCREEN_WIDTH - (line2.length() * 6)) / 2;
    int16_t textY1 = (SCREEN_HEIGHT - 16) / 2; // İlk satır için Y koordinatı
    int16_t textY2 = textY1 + 8; // İkinci satır için Y
koordinatı
    // Çember çiz
    int radius = 31; // Çemberin yarıçapını belirle
    display.drawCircle(SCREEN_WIDTH / 2, SCREEN_HEIGHT / 2, radius, WHITE);
    // Mesajları ekrana yaz
    display.setCursor(textX, textY1);
    display.println(line1);
    display.setCursor(textX, textY2);
    display.println(line2);
    // Ekranı güncelle
    display.display();
}
```

```
Serial.println("Geri sayım için saat, dakika ve saniye girin (Örn: 1:10:30):");
}

void loop() {
    if (Serial.available() > 0) {
        inputString = Serial.readStringUntil('\n');
        inputString.trim();
        int firstColonIndex = inputString.indexOf(':');
        int secondColonIndex = inputString.indexOf(':', firstColonIndex + 1);
        if (firstColonIndex != -1 && secondColonIndex != -1) {
            hours = inputString.substring(0, firstColonIndex).toInt();
            minutes = inputString.substring(firstColonIndex + 1, secondColonIndex).toInt();
            seconds = inputString.substring(secondColonIndex + 1).toInt();
            totalSeconds = (hours * 3600) + (minutes * 60) + seconds;
            Serial.println("Geri sayım başlıyor");
            for (int i = totalSeconds; i >= 0; i--) {
                int displayHours = i / 3600;
                int displayMinutes = (i % 3600) / 60;
                int displaySeconds = i % 60;
                // Ekranı temizle
                display.clearDisplay();

                // Saat, dakika ve saniyeyi string olarak hazırla
                String timeString = (displayHours < 10 ? "0" : "") +
String(displayHours) + ":" + (displayMinutes < 10 ? "0" : "") +
String(displayMinutes) + ":" + (displaySeconds < 10 ? "0" : "") +
String(displaySeconds);
                // Font boyutu ve ekran ortalama hesaplamaları
                display.setTextSize(2); //
Text boyutu
                int16_t centerX = (SCREEN_WIDTH - (timeString.length() * 12)) / 2; //
Metni ortalamak için X koordinatı
                int16_t centerY = (SCREEN_HEIGHT - 16) / 2; //
Metni ortalamak için Y koordinatı
                // Metnin etrafında bir çember çiz
                int radius = (timeString.length() * 6) + 10; // Çemberin yarıçapını
belirle
                display.drawCircle(SCREEN_WIDTH / 2, SCREEN_HEIGHT / 2, radius,
WHITE);
                // Metni ekrana yaz
```

```
display.setCursor(centerX, centerY);
display.print(timeString);
// Ekranı güncelle
display.display();
delay(1000);
}
// Geri sayım tamamlandığında buzzer'ı 3 saniye boyunca çal
digitalWrite(BUZZER_PIN, HIGH);
delay(1000);
digitalWrite(BUZZER_PIN, LOW);
// Geri sayım tamamlandığında OLED ekrana kısaltılmış mesaj yazdır
display.clearDisplay();
display.setTextSize(1); // Text boyutu
// Mesajları ayarla
String line1 = "SAYIM";
String line2 = "BITTI";
String line3 = "YENI";
String line4 = "SURE";
String line5 = "GIRISI";
// Her bir satır için merkezdeki X ve Y koordinatları
int16_t textX = (SCREEN_WIDTH - (line1.length() * 6)) / 2;
int16_t textY1 = (SCREEN_HEIGHT - 32) / 2; // İlk satır için Y
koordinatı
int16_t textY2 = textY1 + 8; // İkinci satır için Y
koordinatı
int16_t textY3 = textY2 + 8; // Üçüncü satır için Y
koordinatı
int16_t textY4 = textY3 + 8; // Dördüncü satır için Y
koordinatı
int16_t textY5 = textY4 + 8; // Beşinci satır için Y
koordinatı
// Çember çiz
int radius = 31; // Çemberin yarıçapını belirle
display.drawCircle(SCREEN_WIDTH / 2, SCREEN_HEIGHT / 2, radius, WHITE);
// Mesajları ekrana yaz
display.setCursor(textX, textY1);
display.println(line1);
display.setCursor(textX, textY2);
display.println(line2);
display.setCursor(textX, textY3);
display.println(line3);
display.setCursor(textX, textY4);
```

```
display.println(line4);
display.setCursor(textX, textY5);
display.println(line5);
// Ekranı güncelle
display.display();
Serial.println("Geri sayım tamamlandı");
Serial.println("Yeni süre girin (Örn: 1:10:30):");
} else {
    Serial.println("Geçersiz giriş. Lütfen saat, dakika ve saniyeyi ':' ile ayırarak girin (Örn: 1:10:30).");
}
}
```

Adım Adım Yapılacaklar:

1. Donanım Kurulumu (Bağlantılar)

Kartı bağlamadan önce kartın kapalı olduğundan emin olun.

Parça	Nereye Bağlanacak?
Buzzer'ın bir ucu	D12 pinine
Buzzer'ın diğer ucu	GND pinine
OLED ekran	I2C pinlerine (SCL → D1, SDA → D0), VCC → 3.3V, GND → GND

2. Yazılım Kurulumu

- Bilgisayarınıza Deneyap Kart IDE'yi kurun.**
 - <https://ide.deneyapkart.org> adresine gidin.
 - Uygun işletim sistemine göre IDE'yi indirin ve kurun.
- Kodları IDE'ye yapıştırın.**
 - Yukarıda paylaşılan kodu eksiksiz olarak IDE'ye yapıştırın.
- Kodu karta yükleyin.**
 - Deneyap Kart 1A'yı USB ile bilgisayara bağlayın.
 - Portu ve kart modelini IDE'den seçin.
 - "Yükle" tuşuna basarak kodu karta yükleyin.

4. Proje Nasıl Çalışır?

Kod karta yüklendikten sonra:

- OLED ekran üzerinde "SÜRE GİRİŞİ" mesajı görünür.
- Seri monitör üzerinden saat, dakika ve saniye girilir. (Örn: 0:01:30 yaz ve Enter'a bas)
- Girilen süre geri sayılır ve OLED ekranda canlı olarak gösterilir.
- Süre bitince:
 - Buzzer **1 saniye boyunca** öter.
 - OLED ekrana "SAYIM BİTTİ YENİ SÜRE GİRİŞİ" mesajı gelir.
- Yeniden süre girilerek aynı işlem tekrar edilebilir.

Hatırlatma:

1.Seri Monitör ayarı:

Seri monitörü açtıktan sonra **baud rate'i 115200** olarak ayarlamayı unutma. Aksi takdirde veri doğru şekilde görünmez.

2.Süre giriş formatı çok önemli:

Sadece şu formatta girilmelidir: saat:dakika:saniye

Yanlış girişlerde uyarı gelir. Örnek geçerli girişler:

- 0:0:10
- 0:02:30
- 1:00:00

3.Bağlantı kontrolü:

Eğer OLED ekran çalışmıyorsa:

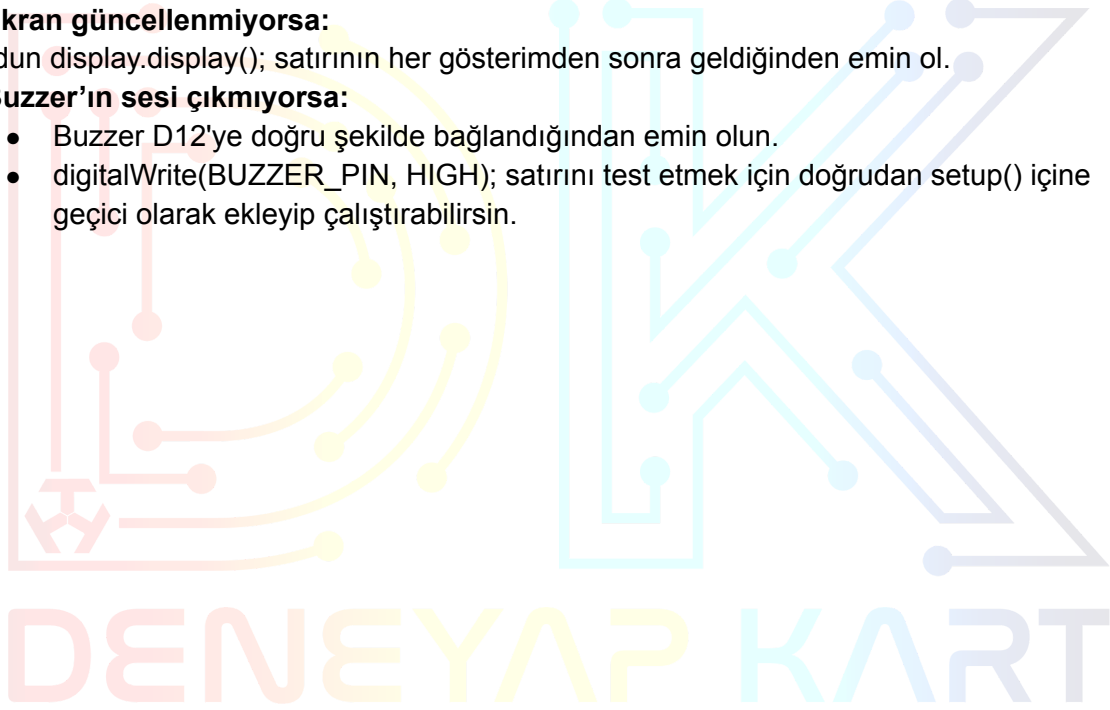
- VCC, GND, SDA, SCL bağlantılarını tekrar kontrol et.
- I2C adresi farklı olabilir, gerekirse 0x3C adresi de denenebilir.

4.Ekran güncellenmiyorsa:

Kodun `display.display();` satırının her gösterimden sonra geldiğinden emin ol.

5.Buzzer'ın sesi çıkmıyorsa:

- Buzzer D12'ye doğru şekilde bağlandığından emin olun.
- `digitalWrite(BUZZER_PIN, HIGH);` satırını test etmek için doğrudan `setup()` içine geçici olarak ekleyip çalıştırabilirsin.



5. Proje Adı: Telegram Tabanlı Uçuş Verisi İzleme Sistemi

Proje Özeti: Deneyap 6-Eksen Ataletsel Ölçüm Birimi ile X, Y, Z eksenlerindeki ivme ve jiro değerleri okunur. Sıcaklık verisi hem °C hem de °F cinsinden ölçülür. Deneyap GPS ve GLONASS Konum Belirleyici sayesinde enlem ve boylam bilgileri elde edilir. Deneyap Basınç Ölçer modülünün sağladığı basınç verisi ile Barometrik İrtifa Denklemi kullanılarak irtifa hesaplanır. Deneyap Ultraviyole Işık Algılayıcı ile ışık şiddeti lux cinsinden ölçülür. Tüm veriler her 5 saniyede bir güncellenerek Telegram botu aracılığıyla belirtilen chat ID'ye iletilir.

Gerekli Malzemeler:

- Deneyap Kart 1A
- Breadboard
- 4 adet I2C bağlantı kablosu
- Deneyap GPS ve GLONASS Konum Belirleyici
- Deneyap Basınç Ölçer
- Deneyap 6-Eksen Ataletsel Ölçüm Birimi
- Deneyap Ultraviyole Işık Algılayıcı
- Power-XtraModel PX103450 Li-Po pil

Proje Adımları:

1.Donanım Kurulumu

Deneyap GPS ve GLONASS Konum Belirleyici, Deneyap Basınç Ölçer, Deneyap 6-Eksen Ataletsel Ölçüm Birimi, Deneyap Ultraviyole Işık Algılayıcı modüllerini I2C haberleşme konnektörü aracılığıyla Deneyap Kart 1A'ya daisy-chain (zincirleme) konfigürasyonunda Proje Devre Şeması adlı başlıkta gösterildiği gibi bağlanmalıdır.

Modüller I2C konnektörleri üzerinden tek hat üzerinden birbirine bağlanarak Deneyap Kart 1A'ya bağlanmalıdır. Her modül I2C protokolü üzerinden benzersiz adreslerle haberleşir.

2.Telegram Bot Oluşturma

1. Aşağıdaki adımları takip ederek bir Telegram botu oluşturunuz:
2. Telegram uygulamasını açın.
3. Arama (lup) sekmesine "BotFather" yazarak aratın.
4. Mavi tikli resmi BotFather botuna giriş yapın.
5. /start komutunu gönderin.
6. Ardından /newbot komutunu göndererek yeni bot oluşturma işlemini başlatın.
7. Botunuza bir isim verin (örnek: Uçuş Verisi İzleme Sistemi).
8. Kullanıcı adı (username) belirleyin (örnek: ucusverisi_bot gibi, "bot" eki zorunludur).
9. BotFather tarafından gönderilen mesajda geçen "Use this token to access the HTTP API" kısmındaki bot token bilgisini saklayın.

3.Telegram Chat ID (Kullanıcı Kimliği) Öğrenme

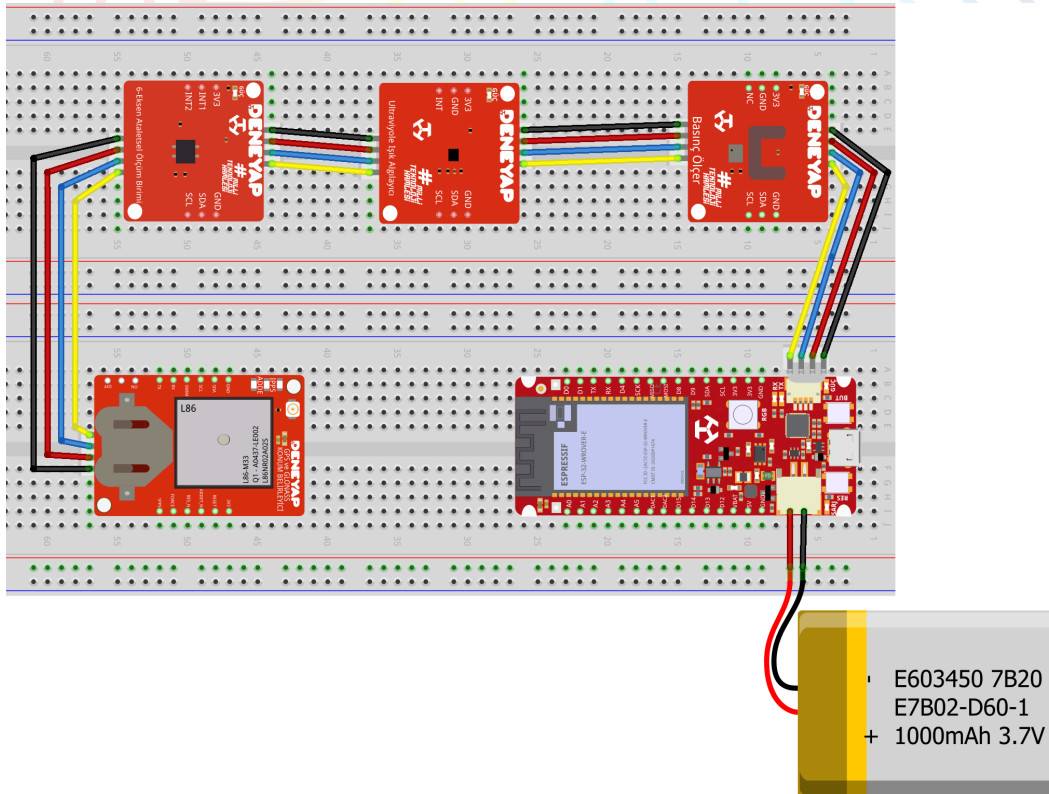
1. Botun mesaj gönderebilmesi için Telegram kullanıcı kimliğinizi (chat ID) öğrenin:
2. Telegram arama çubuğuna @userinfobot yazın ve ilgili botu açın.
3. /start komutunu gönderin.
4. Bot tarafından dönen mesajda yer alan ID değeri sizin kullanıcı kimliğinizdir. Bu değeri kopyalayın ve güvenli bir yerde saklayın.

4.Kod Yükleme

1. Arduino IDE veya Deneyap Kart IDE'yi açın.
2. Deneyap Kart 1A'yı bilgisayara USB kablosu aracılığıyla bağlayın.
3. Kart menüsünden uygun kart modelini ve Port menüsünden bağlanan portu seçin.
4. "Proje Kodları" başlığı altında verilen kaynak kodu IDE'ye yapıştırın.
5. Kod içerisinde aşağıdaki alanları kendi bilgilerinizle güncelleyin:
// WiFi ağınızın adını giriniz.
6. `const char* ssid = "YOUR_WIFI_SSID";`
// WiFi şifrenizi giriniz.
7. `const char* password = "YOUR_WIFI_PASSWORD";`
// Telegram bot tokeninizi giriniz.
8. `#define BOTtoken "YOUR_BOT_TOKEN"`

// Telegram chat kimliğinizi giriniz.
9. `#define CHAT_ID "YOUR_CHAT_ID"`
10. Kodu derleyin ve derleme tamamlandığında Deneyap Kart 1A 'ya yükleyin.

Proje Devre Şeması:



Proje Kodları :

```
#include <Deneyap_6EksenAtaletselOlcumBirimi.h>
#include <Deneyap_GPSveGLONASSkonumBelirleyici.h>
#include <Deneyap_BasincOlcer.h>
#include <Deneyap_UltraviyoleIsikAlgilici.h>
#include <DeneyapTelegram.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>

LSM6DSM AccGyro;
GPS GPS;
AtmosphericPressure BaroSensor;
UVlight UVlight;
// WiFi ağınızın adını giriniz. (Enter your WiFi SSID here.)
const char* ssid = "YOUR_WIFI_SSID";
// WiFi şifrenizi giriniz. (Enter your WiFi password here.)
const char* password = "YOUR_WIFI_PASSWORD";
// Telegram bot tokeninizi giriniz. (Enter your Telegram bot token here.)
#define BOTtoken "YOUR_BOT_TOKEN"
// Telegram chat kimliğinizi giriniz. (Enter your Telegram chat ID here.)
#define CHAT_ID "YOUR_CHAT_ID"
DeneyapTelegram telegram;
WiFiClientSecure client;

void setup() {
    Serial.begin(115200);
    delay(1000);
    AccGyro.begin(0x6B);
    if (!GPS.begin(0x2F)) {
        Serial.println("GPS başlatılamadı. (GPS could not be initialized.)");
        while (1);
    }

    BaroSensor.begin(0x76);
    UVlight.begin(0x53);
    UVlight.setMode(LTR390_MODE_ALS);
    client.setInsecure();
    Serial.print("WiFi'ye bağlanılıyor (Connecting to WiFi): ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
```

```
while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
}
Serial.println("WiFi'ye bağlandı. (WiFi connected.)");
Serial.print("IP address: ");
Serial.println(WiFi.localIP());

telegram.begin(BOTtoken);
telegram.send(CHAT_ID, "Cihaz bağlandı. (Device is Connected.)");
}

void loop() {
    float accelX = AccGyro.readFloatAccelX();
    float accelY = AccGyro.readFloatAccelY();
    float accelZ = AccGyro.readFloatAccelZ();

    float gyroX = AccGyro.readFloatGyroX();
    float gyroY = AccGyro.readFloatGyroY();
    float gyroZ = AccGyro.readFloatGyroZ();

    float tempC = AccGyro.readTempC();
    float tempF = AccGyro.readTempF();

    float lat = GPS.readLocationLat();
    float lng = GPS.readLocationLng();
    float pressure = BaroSensor.getPressure();
    float lux = UVlight.getLUX();

    // Barometrik İrtifa Denklemi (Barometric Altitude Equation)
    float altitude = 44330 * (1.0 - pow(pressure / 1013.25, 0.1903));
    String locationMessage = "Konum (Location): " + String(lat, 7) + ", " +
String(lng, 7);
    telegram.send(CHAT_ID, locationMessage);
    String accelMessage = "İvme (Accel) (X,Y,Z): " + String(accelX) + ", " +
String(accelY) + ", " + String(accelZ);
    telegram.send(CHAT_ID, accelMessage);
    String gyroMessage = "Jiro (Gyro) (X,Y,Z): " + String(gyroX) + ", " +
String(gyroY) + ", " + String(gyroZ);
    telegram.send(CHAT_ID, gyroMessage);
    String tempMessage = "Sıcaklık (Temperature): " + String(tempC) + " °C, " +
String(tempF) + " °F";
```

```
telegram.send(CHAT_ID, tempMessage);  
String pressureMessage = "Basınç (Pressure): " + String(pressure) + " hPa";  
telegram.send(CHAT_ID, pressureMessage);  
String altitudeMessage = "İrtifa (Altitude): " + String(altitude) + "  
meters";  
telegram.send(CHAT_ID, altitudeMessage);  
String luxMessage = "Işık Şiddeti (LUX): " + String(lux);  
telegram.send(CHAT_ID, luxMessage);  
delay(5000);  
}
```

Hatırlatma:

Kod içerisinde bulunan parametrelerin, size ait olan Telegram bot token'ı ve kullanıcıya/sohbete özel chat ID ile değiştirildiğinden emin olun.

Proje GitHub Linki:

https://github.com/KaganxSahin/TelegramBasedFlightDataMonitoringSystem_DeneyapKart1A

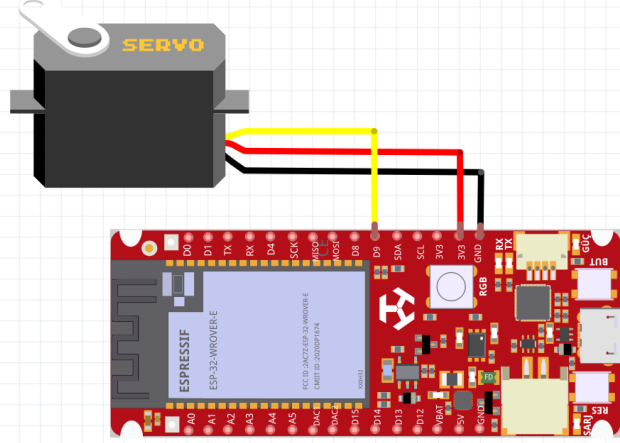
DENEYAP KART

6. Proje Adı: El Hareketiyle Servo Kontrolü AI Projesi

Gerekli Malzemeler:

- 1 Adet Deneyap Kart
- 1 Adet Servo Motor
- 3 Adet Jumper Kablo

Proje Devre Şeması:



Proje Kodları:

Bilgisayarda çalışacak kodlar:

```
import cv2
import mediapipe as mp
import math
import serial # Seri port için pyserial

# Seri portu ayarla (Port adı değiştirilmeli)
ser = serial.Serial('COM10', 9600)

# Mediapipe hazırlığı
mp_hands = mp.solutions.hands
hands = mp_hands.Hands(min_detection_confidence=0.7,
min_tracking_confidence=0.7)
mp_draw = mp.solutions.drawing_utils

# Kamerayı başlat
cap = cv2.VideoCapture(0)
```

```
def calculate_angle(a, b):
    """İki nokta arasındaki doğrusal açı (0-180 derece)."""
    dx = b[0] - a[0]
    dy = b[1] - a[1]
    angle = math.atan2(dy, dx)
    angle = math.degrees(angle)

    # Açıyı pozitif yap ve 0-180 derece arasında sınırlı tut

    return abs(angle)

while True:
    ret, frame = cap.read()
    if not ret:
        break

    # Görüntüyü işlemek için çevir
    frame = cv2.flip(frame, 1) # Ayna görüntüsü
    image = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)
    results = hands.process(image)

    if results.multi_hand_landmarks:
        for hand_landmarks in results.multi_hand_landmarks:
            landmarks = hand_landmarks.landmark
            points = [(int(lm.x * frame.shape[1]), int(lm.y * frame.shape[0]))
for lm in landmarks]

            # İşaret parmağı ve baş parmak
            thumb_tip = points[4]
            index_tip = points[8]

            # Başparmak ve işaret parmağı arasındaki açı
            angle = calculate_angle(thumb_tip, index_tip)

            # Seri porta yazdır
            ser.write(f"{int(angle)}\n".encode('utf-8')) # Açı değeri tam
sayı olarak gönderilir

            # Çizgiyi çiz
            cv2.line(frame, thumb_tip, index_tip, (0, 255, 0), 3)
            # ekrana açı bilgisini yaz
```

```
cv2.putText(frame, str(angle), (20,80), cv2.FONT_HERSHEY_PLAIN, 2,
(255,0,0),3)
# Landmark'ları çiz
mp_draw.draw_landmarks(frame, hand_landmarks,
mp_hands.HAND_CONNECTIONS)

cv2.imshow("Hand Gesture", frame)

# 'q' veya 'Esc' ile çıkış
if cv2.waitKey(1) & 0xFF in [ord('q'), 27]:
    break

cap.release()
ser.close() # Seri portu kapat
cv2.destroyAllWindows() # Ekranı kapat
```

Proje Kodları:

Deneyap Kart üzerinde çalışacak kodlar:

```
#include <Deneyap_Servo.h>

Servo myServo;
int angle; // Seri porttan alınan açı

void setup() {

    Serial.begin(9600); // Seri haberleşme hızı
    myServo.attach(D9);
    myServo.write(90); // Başlangıç açısını 90 derece
}

void loop() {
    if (Serial.available() > 0) { // Seri portta veri varsa
        String data = Serial.readStringUntil('\n'); // Satır sonuna kadar oku
        angle = data.toInt(); // Açı değerini tamsayıya
        // çevir
        angle = constrain(angle, 0, 180); // Açıyı 0-180 derece
        // aralığında sınırlı tut
        myServo.write(angle); // Servo motoru açıya göre
        // döndür
    }
}
```

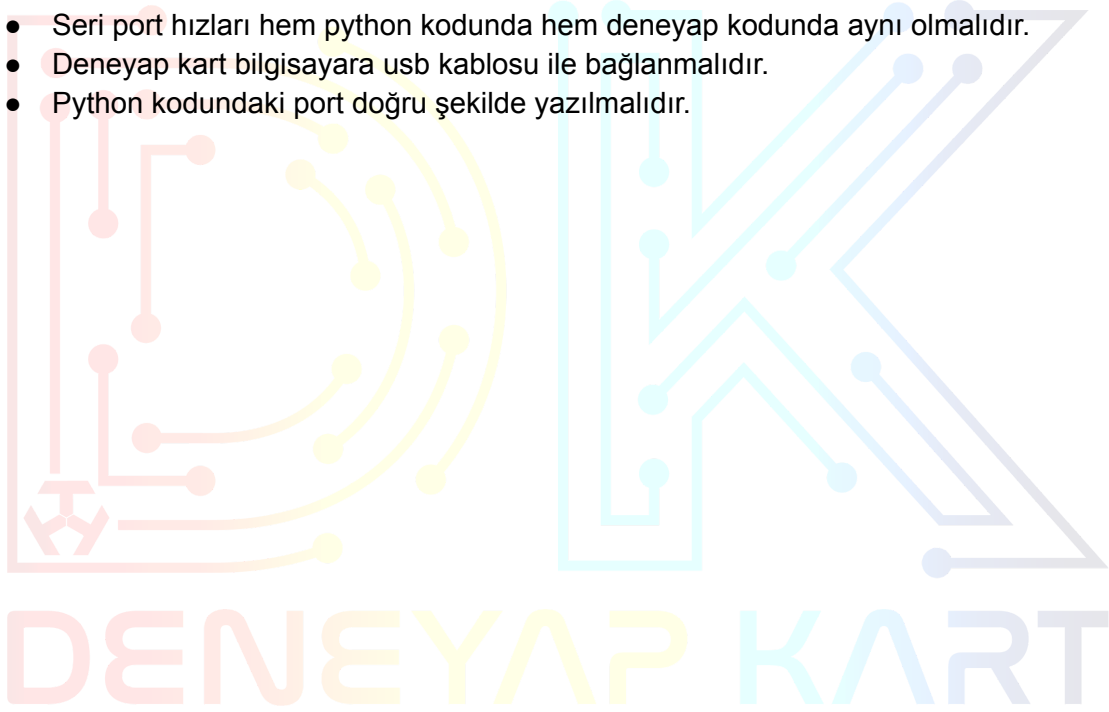
```
}  
}
```

Yapılması Gereken Adımlar:

- Bilgisayara Python yüklenmelidir.
- Daha sonra python üzerinde çalışacak ve projede kullanılacak opencv, mediapipe ve pyserial kütüphanelerli **pip** paket yükleyicisiyle yüklenmelidir.
- Deneyap Kart IDE'den servo kütüphanesi yüklenmelidir.
- Doğru kart ve port seçerek kod karta yüklenmelidir.
- Daha sonra ise python kodu deneyap kart bilgisayara bağlı iken ve doğru port kodda yazılmışken çalıştırılır.

Hatırlatma:

- Seri port hızları hem python kodunda hem deneyap kodunda aynı olmalıdır.
- Deneyap kart bilgisayara usb kablosu ile bağlanmalıdır.
- Python kodundaki port doğru şekilde yazılmalıdır.



7. Proje Adı: Çevresel Güvenlik ve Uyarı Sistemi

Proje Özeti: MQ-2 Gaz Sensörü havadaki gaz yoğunluğunu ölçerek analog veri üretir. Deneyap Basınç Ölçer modülü ise hava basıncını (hPa cinsinden) ve sıcaklığı (°C cinsinden) belirler. Tüm veriler her saniye güncellenerek seri port ekranında görüntülenir. Ölçülen değerlerin belirlenen alt ve üst eşiklerin dışına çıkması durumunda Telegram botu aracılığıyla ilgili chat ID'ye bildirim gönderilir ve buzzer aktif hale getirilerek sesli uyarı verilir.

Gerekli Malzemeler:

- 1 Adet Deneyap Kart 1A
- 1 Adet Breadboard
- 1 Adet Deneyap Basınç Ölçer
- 1 Adet Aktif Buzzer
- 1 Adet MQ-2 Gaz Sensörü
- 1 Adet I2C bağlantı kablosu
- 6 Adet jumper kablo
- 1 Adet Power-XtraModel PX103450 Li-Po pil

Proje Adımları:

1. Donanım Kurulumu

MQ-2 gaz sensörünün bağlantıları yapılırken, sensörün besleme gerilimi olan VCC pini Deneyap Kart 1A'nın 5V pinine, topraklama (GND) pini ise kart üzerindeki GND pinine bağlanmalıdır. Sensörden analog veri okunabilmesi için AOUT (analog çıkış) pini, Deneyap Kart 1A üzerindeki A0 pinine bağlanarak sensörün algıladığı gaz yoğunluğunun analog sinyal olarak karta iletilmesi sağlanır. Aktif buzzer bağlantısında, sesli uyarı sisteminin doğru çalışabilmesi için buzzerin pozitif (+) pini Deneyap Kart 1A üzerindeki dijital D0 pinine, negatif (-) pini ise GND pinine bağlanmalıdır. Bu bağlantı sayesinde kart, belirli bir eşik değeri aşıldığında buzzer üzerinden sesli alarm verebilir. Son olarak Deneyap Basınç Ölçer, I2C haberleşme protokolünü kullanarak Deneyap Kart 1A'ya bağlanır.

2. Telegram Bot Oluşturma

1. Aşağıdaki adımları takip ederek bir Telegram botu oluşturunuz:
2. Telegram uygulamasını açın.
3. Arama (lup) sekmesine "BotFather" yazarak aratın.
4. Mavi tikli resmi BotFather botuna giriş yapın.
5. /start komutunu gönderin.
6. Ardından /newbot komutunu göndererek yeni bot oluşturma işlemini başlatın.
7. Botunuza bir isim verin (örnek: Çevresel Güvenlik ve Uyarı Sistemi).
8. Kullanıcı adı (username) belirleyin (örnek: EnvironmentalSafetyAndAlert_bot gibi, "bot" eki zorunludur).
9. BotFather tarafından gönderilen mesajda geçen "Use this token to access the HTTP API" kısmındaki bot token bilgisini saklayın.

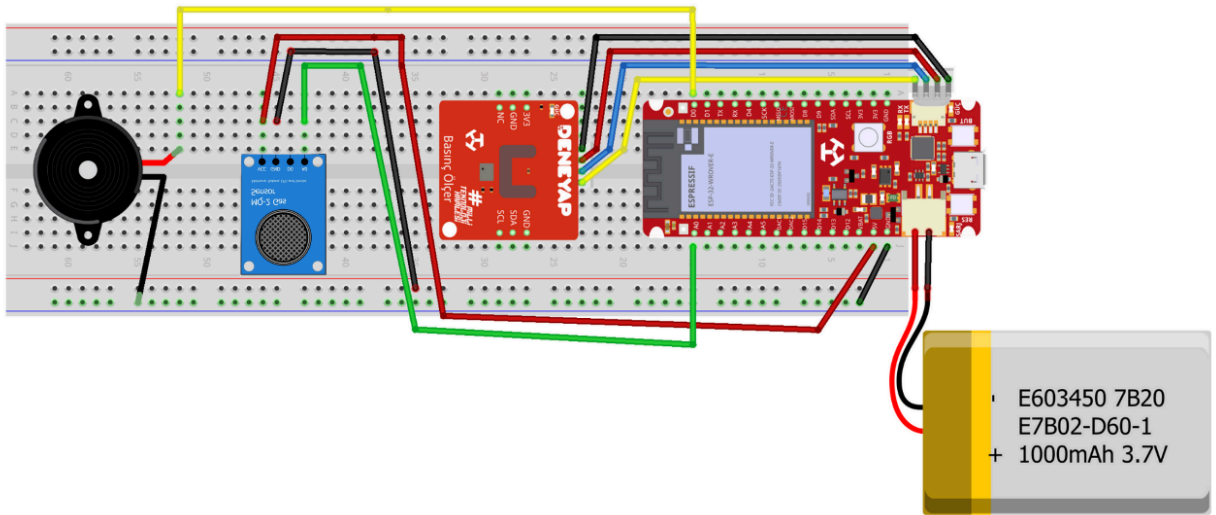
3.Telegram Chat ID (Kullanıcı Kimliği) Öğrenme

1. Botun mesaj gönderebilmesi için Telegram kullanıcı kimliğinizi (chat ID) öğrenin:
2. Telegram arama çubuğuna @userinfobot yazın ve ilgili botu açın.
3. /start komutunu gönderin.
4. Bot tarafından dönen mesajda yer alan ID değeri sizin kullanıcı kimliğinizdir. Bu değeri kopyalayın ve güvenli bir yerde saklayın.

4.Kod Yükleme

1. Arduino IDE veya Deneyap Kart IDE'yi açın.
2. Deneyap Kart 1A'yı bilgisayara USB kablosu aracılığıyla bağlayın.
3. Kart menüsünden uygun kart modelini ve Port menüsünden bağlanan portu seçin.
4. "Proje Kodları" başlığı altında verilen kaynak kodu IDE'ye yapıştırın.
5. Kod içerisinde aşağıdaki alanları kendi bilgilerinizle güncelleyin:
// WiFi ağınızın adını giriniz. (Enter your WiFi SSID here.)
6. const char* ssid = "YOUR_WIFI_SSID";
// WiFi şifrenizi giriniz. (Enter your WiFi password here.)
7. const char* password = "YOUR_WIFI_PASSWORD";
// Telegram bot tokeninizi giriniz. (Enter your Telegram bot token here.)
8. #define BOTtoken "YOUR_BOT_TOKEN"
// Telegram chat kimliğinizi giriniz. (Enter your Telegram chat ID here.)
9. #define CHAT_ID "YOUR_CHAT_ID"
10. Kodu derleyin ve derleme tamamlandığında Deneyap Kart 1A 'ya yükleyin.

Proje Devre Şeması:



Proje Kodları :

```
#include <Deneyap_BasincOlcer.h>
#include <DeneyapTelegram.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>

AtmosphericPressure BaroSensor;
DeneyapTelegram telegram;
WiFiClientSecure client;

// Wi-Fi ağınızın adını giriniz. (Enter your Wi-Fi SSID here.)
const char* ssid = "YOUR_WIFI_SSID";
// Wi-Fi şifrenizi giriniz. (Enter your Wi-Fi password here.)
const char* password = "YOUR_WIFI_PASSWORD";
// Telegram bot tokeninizi giriniz. (Enter your Telegram bot token here.)
#define BOTtoken "YOUR_BOT_TOKEN"
// Telegram chat kimliğinizi giriniz. (Enter your Telegram chat ID here.)
#define CHAT_ID "YOUR_CHAT_ID"

const float maxTemperature = 50.0;
const float minTemperature = -10.0;

const float maxPressure = 1600.0;
const float minPressure = 600.0;

const int maxGasAnalog = 2000;
const int minGasAnalog = 2;

#define buzzerPin D0
#define MQ2_ANALOG_PIN A0

void setup() {
    Serial.begin(115200);
    delay(1000);

    BaroSensor.begin(0x76);
    pinMode(buzzerPin, OUTPUT);

    client.setInsecure();
    Serial.print("Wi-Fi'ye bağlanılıyor: (Connecting to Wi-Fi: )");
```

```
Serial.println(ssid);
WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
}
Serial.println("WiFi'ye bağlandı. (Connected to WiFi.)");
Serial.print("IP adresi: (IP Address: )");
Serial.println(WiFi.localIP());
telegram.begin(BOTtoken);
telegram.send(CHAT_ID, "Cihaz bağlandı. (Device connected.)");
pinMode(buzzerPin, OUTPUT);
pinMode(MQ2_ANALOG_PIN, INPUT);
}

void loop() {
    float pressure = BaroSensor.getPressure();
    float temperature = BaroSensor.getTemp();
    int gasAnalog = analogRead(MQ2_ANALOG_PIN);
    Serial.print("Basınç: (Pressure: )");
    Serial.println(pressure);
    Serial.print("Sıcaklık: (Temperature: )");
    Serial.println(temperature);
    Serial.print("Gaz Seviyesi: (Gas Level: )");
    Serial.println(gasAnalog);

    if (temperature > maxTemperature) {
        telegram.send(CHAT_ID, "Sıcaklık tehlikeli seviyeye ulaştı. (Temperature reached dangerous level.)");
        String tempMessage = "Sıcaklık (Temperature): " + String(temperature) + " °C";
        telegram.send(CHAT_ID, tempMessage);
        digitalWrite(buzzerPin, HIGH);
    } else if (temperature < minTemperature) {
        telegram.send(CHAT_ID, "Sıcaklık tehlikeli seviyeye düştü. (Temperature dropped to dangerous level.)");
        String tempMessage = "Sıcaklık (Temperature): " + String(temperature) + " °C";
        telegram.send(CHAT_ID, tempMessage);
        digitalWrite(buzzerPin, HIGH);
    }

    if (pressure > maxPressure) {
```

```
telegram.send(CHAT_ID, "Basınç tehlikeli seviyeye ulaştı. (Pressure
reached dangerous level.)");
String pressureMessage = "Basınç (Pressure): " + String(pressure) + "
hPa";
telegram.send(CHAT_ID, pressureMessage);

digitalWrite(buzzerPin, HIGH);
} else if (pressure < minPressure) {
    telegram.send(CHAT_ID, "Basınç tehlikeli seviyeye düştü. (Pressure dropped
to dangerous level.)");
    String pressureMessage = "Basınç (Pressure): " + String(pressure) + "
hPa";

    telegram.send(CHAT_ID, pressureMessage);
    digitalWrite(buzzerPin, HIGH);
}
if (gasAnalog > maxGasAnalog) {
    telegram.send(CHAT_ID, "Gaz tehlikeli seviyeye ulaştı. (Gas reached
dangerous level.)");
    String gasMessage = "Gaz seviyesi (Gas level): " + String(gasAnalog);
    telegram.send(CHAT_ID, gasMessage);
    digitalWrite(buzzerPin, HIGH);
} else if (gasAnalog < minGasAnalog) {
    telegram.send(CHAT_ID, "Gaz tehlikeli seviyeye düştü. (Gas dropped to
dangerous level.)");
    String gasMessage = "Gaz seviyesi (Gas level): " + String(gasAnalog);
    telegram.send(CHAT_ID, gasMessage);
    digitalWrite(buzzerPin, HIGH);
}
if (temperature >= minTemperature && temperature <= maxTemperature &&
pressure >= minPressure && pressure <= maxPressure && gasAnalog >=
minGasAnalog && gasAnalog <= maxGasAnalog) {
    digitalWrite(buzzerPin, LOW);
}
delay(1000);
}
```

Hatırlatma:

Kod içerisinde bulunan parametrelerin, size ait olan Telegram bot token'ı ve kullanıcıya/sohbete özel chat ID ile değiştirildiğinden emin olun.

Belirtilen eşik değerler, sistem tasarımına ve çevresel koşullara göre kullanıcı tarafından serbestçe belirlenmelidir. Örneğin maxTemperature, sistemin güvenli çalışabileceği en yüksek sıcaklık seviyesini ifade eder.

```
const float maxTemperature = 50.0; // Maksimum kabul edilen sıcaklık değeri (°C)
const float minTemperature = -10.0; // Minimum kabul edilen sıcaklık değeri (°C)
const float maxPressure = 1600.0; // Maksimum kabul edilen basınç değeri (hPa)
const float minPressure = 600.0; // Minimum kabul edilen basınç değeri (hPa)
const int maxGasAnalog = 2000; // Gaz sensöründen alınan maksimum analog değer
const int minGasAnalog = 2; // Gaz sensöründen alınan minimum analog değer
```

Proje GitHub Linki:

https://github.com/KaganxSahin/EnvironmentalSafetyAndAlertSystem_Deneyap_Kart1A

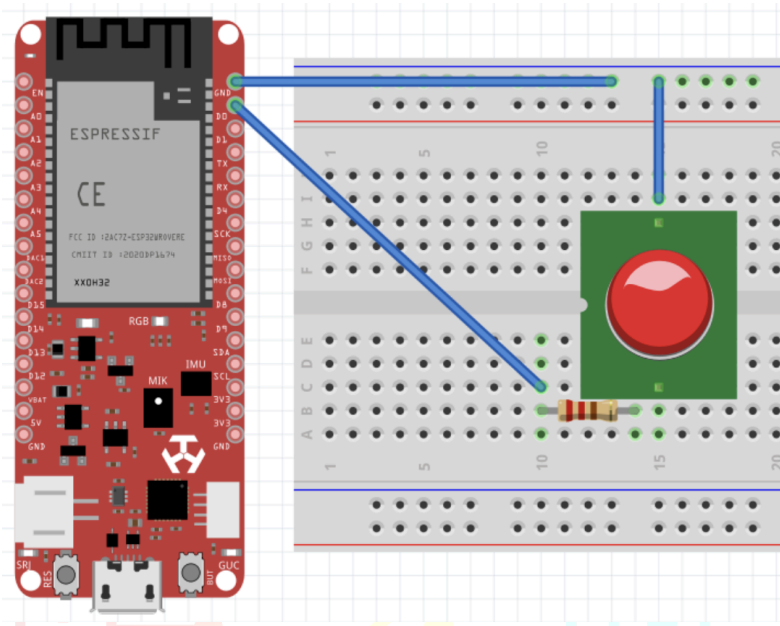


8. Proje Adı: Deneyap Kart ile Melodi Çalma

Gerekli Malzemeler:

- 1 Adet Deneyap Kart
- 1 Adet Buzzer
- 1 adet 220 ohm Direnç
- Jumper Kablolar

Proje Devre Şeması:



1. Buzzer'ın + bacağı → Deneyap Kart D31 pinine
2. Buzzer'ın - bacağı → Direnç → GND (toprak hattı)

 Not: Eğer pasif buzzer kullanıyorsan frekansla farklı notalar çalınabilir. Aktif buzzer kullanıyorsan sadece "bip" sesi duyarsın ve müzik çalma pek mümkün olmaz

Proje Kodları:

```
#define sol 392
#define la 440
#define fa 349
#define mi 329
#define re 293
#define sib 466
#define dod 277
#define do 261
int buzzerPin=31;
```

```
int nota
[]={la,re,fa,mi,re,fa,re,mi,re,sib,do,la,la,re,fa,mi,re,fa,re,mi,re,sib,la,sol,
sol,sib,dod,mi,sol,sib,dod,re,re,fa,do,sib,la,do,sib,sib,la,la,dod,re,la,re,f
a,mi,re,fa,re,mi,re,sib,do,la,la,re,fa,mi,re,fa,re,mi,re,sib,la};

int sure
[]={500,500,500,500,500,500,500,500,500,500,500,2500,500,500,500,500,500,5
00,500,500,500,500,2500,500,500,500,2500,500,500,500,2500,500,500,500,500,
500,500,500,500,500,2500,500,500,500,500,500,500,500,500,500,500,2500,
500,500,500,500,500,500,500,500,500};

void setup(){
pinMode(buzzerPin,OUTPUT);

}

void loop(){
for (int i=0;i<67;i++){
tone (buzzerPin,nota[i]);
delay(sure[i]);
}
}
```

Adım Adım Yapılacaklar:

1. Donanımı Kur

- Buzzer'ı doğru pinlere bağla (D31 ve GND'ye dirençle).
- Bağlantıların gevşek olmamasına dikkat et.

2. Deneyap IDE'yi Aç ve Kodları Yapıştır

- IDE: <https://ide.deneyapkart.org>
- Kart bağlantısını seç ve kodu yapıştır.

3. Kodu Karta Yükle

- Bağlan butonuyla karta bağlan.
- Yükle diyerek kodu gönder.

4. Sonucu Dinle

- Buzzer'dan melodinin çaldığını duymalısın.
- Eğer ses çıkmıyorsa:
 - Buzzer tipi doğru mu (aktif/pasif)?
 - Bağlantılar doğru mu?
 - Kod karta düzgün yüklendi mi?

İpuçları:

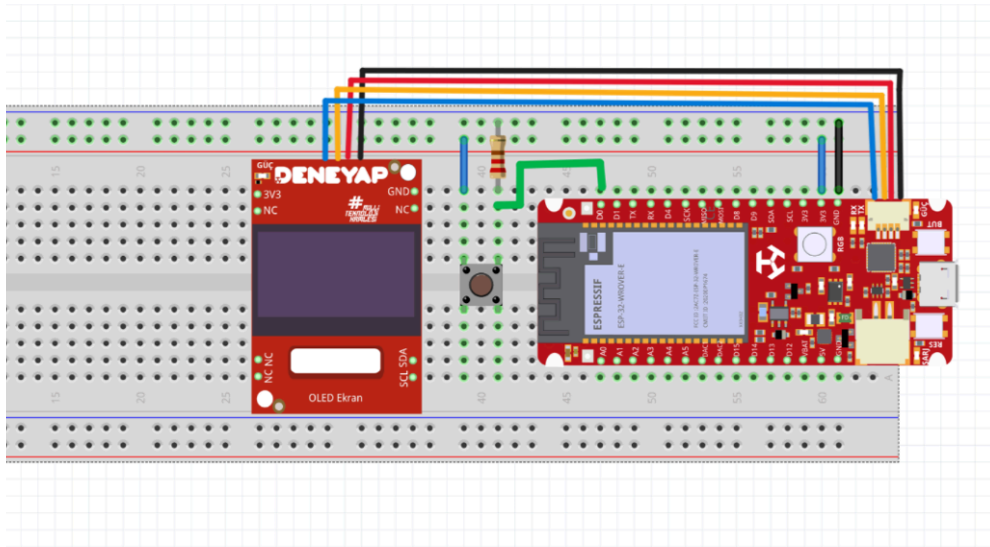
- Melodiyi sonsuz tekrar ettirmek istersen loop() içine while(true){...} ile koyabilirsin.
- Farklı melodiler denemek istersen yeni nota[] ve sure[] dizileri yazabilirsin.
- Her notadan sonra noTone(buzzerPin); eklemek sesi netleştirir.

9. Proje Adı: OLED ile Google Dinozor Simülasyon Oyunu

Gerekli Malzemeler:

- 1 Adet Deneyap Kart 1A
- 1 Adet Deneyap OLED Ekran
- 1 Adet Buton
- 1 Adet 220 ohm Direnç
- 4 Adet Jumper Kablo
- 1 Adet I2C Bağlantı Kablosu

Proje Devre Şeması:



Proje Kodları :

```
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define SCREEN_WIDTH 128
#define SCREEN_HEIGHT 64
#define OLED_RESET -1
Adafruit_SSD1306 display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);

const int buttonPin = D0;
int buttonState = 0;
bool isJumping = false; // başlangıçtaki değer
int dinoY = 34; // Çizginin üstüne hizalama (konum belirleme)
int velocity = 0;
int maxObstacles = 3; // ekrana çıkacak maksimum engel sayısı
int obstacleWidth = 10;
int gap = 50; // Engel grupları arasındaki boşluk
```



```
struct Obstacle { //engel class tanımlaması
    int x;           //ekranda kaydırma konumu
    bool isActive;
};

Obstacle obstacles[3]; // Dizinin boyutu maxObstacles'a göre ayarlandı yani
maksimum 3 engel aynı anda gönderilebilir
int score = 0;
bool gameOver = false;
void setup() {
    pinMode(buttonPin, INPUT);
    if (!display.begin(SSD1306_SWITCHCAPVCC, 0x3D)) { //oled ekran başlatma ve
oled ekranı adresine ulaşma, bu adres 0x3D veya 0x3C dir.
        Serial.println(F("SSD1306 allocation failed"));
        for (;;);
    }
    display.display();
    delay(2000);
    display.clearDisplay();
    Serial.begin(115200);
    // Engelleri başlangıç pozisyonuna yerleştir
    for (int i = 0; i < maxObstacles; i++) {
        obstacles[i].x = SCREEN_WIDTH + i * (obstacleWidth + gap);
        obstacles[i].isActive = true;
    }
}

void loop() {
    if (!gameOver) {
        buttonState = digitalRead(buttonPin);
        if (buttonState == HIGH && !isJumping) {
            isJumping = true; //butona basıldığında dino zıplar
            velocity = -10;
        }
        if (isJumping) {
            dinoY += velocity;
            velocity += 1;
            if (dinoY >= 34) { // Çizginin üstüne inme kontrolü
                dinoY = 34;
                isJumping = false;
                velocity = 0;
            }
        }
    }
}
```

```
// Engelleri hareket ettir
for (int i = 0; i < maxObstacles; i++) {
    if (obstacles[i].isActive) {
        obstacles[i].x -= 3; // Engellerin daha hızlı hareket edebilmesi için
        // bu sayıyı arttırabilirsiniz
        if (obstacles[i].x + obstacleWidth < 0) {
            obstacles[i].x = SCREEN_WIDTH + random(0, maxObstacles) *
            (obstacleWidth + gap); // engellerin kaç tane ve nereden geleceği burada
            belirtilir
            obstacles[i].isActive = false;
        }
        // Dinozor ile çarpışma kontrolü
        if (obstacles[i].isActive && obstacles[i].x <= 25 && obstacles[i].x +
        obstacleWidth >= 10 && dinoY + 20 >= 34) { // dino ile aynı koordinatlar
        olduğunda
            gameOver = true;
        }
        // oyunu bitir
    }
}

// Yeni engel ekleme
for (int i = 0; i < maxObstacles; i++) {
    if (!obstacles[i].isActive) {
        obstacles[i].x = SCREEN_WIDTH + random(0, maxObstacles) *
        (obstacleWidth + gap);
        obstacles[i].isActive = true;
        score++; // Her yeni engelde skoru artır
    }
}
} else {
    if (digitalRead(buttonPin) == HIGH) { // oyun bittiğinde butona basılır
    ve baştan başlanır
        resetGame();
    }
}

display.clearDisplay();
drawGround();
drawDino(dinoY);
for (int i = 0; i < maxObstacles; i++) {
    if (obstacles[i].isActive) {
        drawObstacle(obstacles[i].x); // belirlenen koordinatlara engel çizilir
        ve hareket esnasında korunması sağlanır
    }
}
```

```
    }  
  }  
  drawScore();  
  if (gameOver) {  
    drawGameOver();  
  }  
  display.display();  
  delay(50);  
}  
  
void drawGround() {  
  display.drawLine(0, 50, 128, 50, SSD1306_WHITE); // Çizgiyi 50'ye  
  konumlandırdık  
}  
  
void drawDino(int y) {  
  display.fillRect(10, y, 15, 20, SSD1306_WHITE); // Dinozorun boyu 20,  
  genişliği 15 olarak belirlendi ve fillRect() ile içi dolu cisim haline geldi  
}  
  
void drawObstacle(int x) {  
  display.fillRect(x, 40, obstacleWidth, 10, SSD1306_WHITE); // Engeli 40'a  
  hizaladık  
}  
  
void drawScore() { //skor yazdırma  
  display.setTextSize(1);  
  display.setTextColor(SSD1306_WHITE);  
  display.setCursor(70, 0);  
  display.print("Score: ");  
  display.println(score);  
}  
  
void drawGameOver() { //game over ekranı  
  display.setTextSize(2);  
  display.setTextColor(SSD1306_WHITE);  
  display.setCursor(20, 20);  
  display.println("Game Over");  
  display.setTextSize(1);  
  display.setCursor(35, 40);  
  display.print("Score: ");  
  display.println(score);  
}
```

```
}  
  
void resetGame() {  
    score = 0; // oyun bitince skor ve durumlar sıfırlanır  
    dinoY = 34;  
    velocity = 0;  
    isJumping = false;  
    gameOver = false;  
    for (int i = 0; i < maxObstacles; i++) { // Engelleri başlangıç pozisyonuna  
        yerleştir  
        obstacles[i].x = SCREEN_WIDTH + i * (obstacleWidth + gap);  
        obstacles[i].isActive = true;  
    }  
}
```

Kodun Özeti:

- **Dinozor:** fillRect() ile çiziliyor, sabit konumda.
- **Zıplama:** Butona basıldığında yukarı fırlar, sonra aşağı iner (yerçekimi efekti).
- **Engeller:** Ekranın sağından sola doğru hareket eder.
- **Çarpışma:** Dinozor ve engel aynı hizadaysa oyun biter.
- **Skor:** Geçilen her engel için artar.
- **Game Over:** Ekranda yazı çıkar, butonla yeniden başlatılır.

Bağlantıları Şeması (Devre Şeması):

- OLED Ekran:
 - VCC → 3.3V
 - GND → GND
 - SDA → SDA (D4)
 - SCL → SCL (D5)
- Buton:
 - Bir ucu → D0 pinine
 - Diğer ucu → GND'ye
 - D0 ile 3.3V arasına 220Ω direnç

 Bu bağlantılar genelde sabittir ama Deneyap kart pinlerine göre küçük değişiklikler olabilir.

Kurulum ve Kod Yükleme:

1. Deneyap Kart IDE'yi indir:
<https://ide.deneyapkart.org/>
2. Gerekli kütüphaneler:
 - Adafruit_GFX.h
 - Adafruit_SSD1306.h
3. Kart bağlantısını yap ve kodu yükle.

Adım Adım Yapılacaklar:

Adım 1: Donanımı Kur

- OLED ve butonu Deneyap Kart'a bağla.
- Direnci doğru bağladığından emin ol.

Adım 2: Kodu IDE'ye Yapıştır

- Yukarıda verdiğim tam kodu kullan.
- Serial.begin(115200); kısmı kartın çalıştığını kontrol etmek için seri portu açar.

Adım 3: Kütüphaneleri Yükle

- Adafruit_SSD1306 ve Adafruit_GFX kütüphanelerini yükle.
- Deneyap IDE kullanıyorsan otomatik gelir, Arduino IDE'de manuel eklemen gerekebilir.

Adım 4: Kodu Karta Yükle

- Doğru kart ve port seçili mi kontrol et.
- Yüledikten sonra ekranın geldiğini ve butona bastığında dinozorun zıpladığını kontrol et.

Adım 5: Oyunu Test Et

- Dinozoru zıplat, engelden kaçır, skoru takip et.
- Çarpınca "Game Over" yazısını görmelisin.
- Yeniden başlatmak için butona tekrar bas.

Hatırlatma:

- OLED ekran I2C protokolüyle çalıştığı için sadece 2 veri hattı yeterli.
- Engeller Obstacle isimli bir yapı (struct) ile tanımlanıyor. Ekranda 3 taneye kadar engel olabilir.
- Fiziksel efekt için "yerçekimi" gibi basit bir mantık var (velocity ve dinoY).

Ek Linkler:

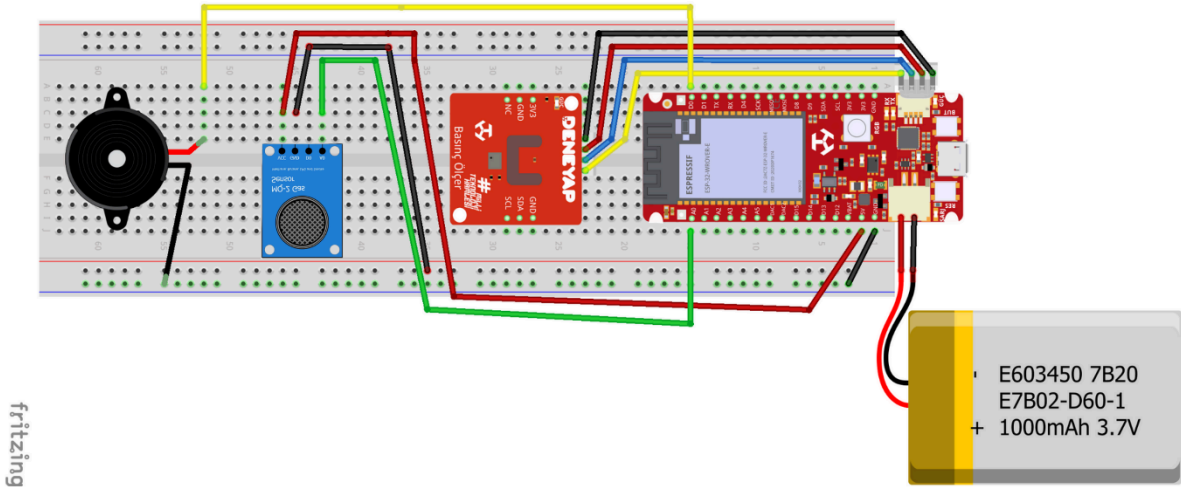
- https://github.com/adafruit/Adafruit_SSD1306
- <https://github.com/adafruit/Adafruit-GFX-Library>
- <https://ide.deneyapkart.org>

10. Proje Adı: Çevresel Güvenlik ve Uyarı Sistemi

Gerekli Malzemeler:

- 1 Adet Deneyap Kart 1A
- 1 Adet Buzzer
- 1 Adet Gaz Sensörü
- 1 Adet Deneyap Basınç Ölçer
- 1 Adet Li-Po Pil
- 6 Adet Jumper Kablo
- 1 Adet I2C Bağlantı Kablosu

Proje Devre Şeması:



//Wiring

/*

Türkçe

- ****MQ-2 Gaz Sensörü****:
 - VCC pinini Deneyap Kart 1A'nın 5V pinine bağlayın.
 - GND pinini Deneyap Kart 1A'nın GND pinine bağlayın.
 - AOOUT (Analog Çıkış) pinini, Deneyap Kart 1A'nın A0 pinine bağlayın.
- ****Aktif Buzzer****:
 - Buzzerın + pinini Deneyap Kart 1A'nın D0 pinine bağlayın.
 - Buzzerın - pinini GND pinine bağlayın.
- ****Deneyap Basınç Ölçer****:
 - I2C haberleşme konnektörü üzerinden Deneyap Kart 1A'ya zincirleme (daisy-chain) bağlantı yaparak bağlayın.

English

- ****MQ-2 Gas Sensor****:

- Connect the VCC pin to the 5V pin of the Deneyap Kart 1A.
 - Connect the GND pin to the GND pin of the Deneyap Kart 1A.
 - Connect the AOUT (Analog Output) pin to the A0 pin of the Deneyap Kart 1A.
 - ****Active Buzzer****:
 - Connect the + pin of the buzzer to the D0 pin of the Deneyap Kart 1A.
 - Connect the - pin of the buzzer to the GND pin.
 - ****Deneyap Pressure Measurer****:
 - Connect in a daisy-chain configuration to Deneyap Kart 1A via the I2C communication connector.
- */

Proje Kodları:

```
#include <Deneyap_BasincOlcer.h>
#include <DeneyapTelegram.h>
#include <WiFi.h>
#include <WiFiClientSecure.h>

AtmosphericPressure BaroSensor;
DeneyapTelegram telegram;
WiFiClientSecure client;

// WiFi ağınızın adını giriniz. (Enter your WiFi SSID here.)
const char* ssid = "YOUR_WIFI_SSID";
// WiFi şifrenizi giriniz. (Enter your WiFi password here.)
const char* password = "YOUR_WIFI_PASSWORD";
// Telegram bot tokeninizi giriniz. (Enter your Telegram bot token here.)
#define BOTtoken "YOUR_BOT_TOKEN"
// Telegram chat kimliğinizi giriniz. (Enter your Telegram chat ID here.)
#define CHAT_ID "YOUR_CHAT_ID"

const float maxTemperature = 50.0;
const float minTemperature = -10.0;

const float maxPressure = 1600.0;
const float minPressure = 600.0;

const int maxGasAnalog = 2000;
const int minGasAnalog = 50;

#define buzzerPin D0
#define MQ2_ANALOG_PIN A0
```

```
void setup() {
  Serial.begin(115200);
  delay(1000);

  BaroSensor.begin(0x76);
  pinMode(buzzerPin, OUTPUT);
  client.setInsecure();
  Serial.print("WiFi'ye bağlanılıyor: (Connecting to WiFi: )");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println("WiFi'ye bağlandı. (Connected to WiFi.)");
  Serial.print("IP adresi: (IP Address: )");
  Serial.println(WiFi.localIP());
  telegram.begin(BOTtoken);
  telegram.send(CHAT_ID, "Cihaz bağlandı. (Device connected.)");
  pinMode(buzzerPin, OUTPUT);
  pinMode(MQ2_ANALOG_PIN, INPUT);
}

void loop() {
  float pressure = BaroSensor.getPressure();
  float temperature = BaroSensor.getTemp();
  int gasAnalog = analogRead(MQ2_ANALOG_PIN);
  Serial.print("Basınç: (Pressure: )");
  Serial.println(pressure);
  Serial.print("Sıcaklık: (Temperature: )");
  Serial.println(temperature);
  Serial.print("Gaz Seviyesi: (Gas Level: )");
  Serial.println(gasAnalog);
  if (temperature > maxTemperature) {
    telegram.send(CHAT_ID, "Sıcaklık tehlikeli seviyeye ulaştı. (Temperature reached dangerous level.)");
    String tempMessage = "Sıcaklık (Temperature): " + String(temperature) + " °C";
    telegram.send(CHAT_ID, tempMessage);
    digitalWrite(buzzerPin, HIGH);
  } else if (temperature < minTemperature) {
```



```
telegram.send(CHAT_ID, "Sıcaklık tehlikeli seviyeye düştü. (Temperature
dropped to dangerous level.)");
String tempMessage = "Sıcaklık (Temperature): " + String(temperature) + "
°C";
telegram.send(CHAT_ID, tempMessage);
digitalWrite(buzzerPin, HIGH);
}
if (pressure > maxPressure) {
    telegram.send(CHAT_ID, "Basınç tehlikeli seviyeye ulaştı. (Pressure
reached dangerous level.)");
    String pressureMessage = "Basınç (Pressure): " + String(pressure) + "
hPa";
    telegram.send(CHAT_ID, pressureMessage);
    digitalWrite(buzzerPin, HIGH);
} else if (pressure < minPressure) {
    telegram.send(CHAT_ID, "Basınç tehlikeli seviyeye düştü. (Pressure dropped
to dangerous level.)");
    String pressureMessage = "Basınç (Pressure): " + String(pressure) + "
hPa";
    telegram.send(CHAT_ID, pressureMessage);
    digitalWrite(buzzerPin, HIGH);
}
if (gasAnalog > maxGasAnalog) {
    telegram.send(CHAT_ID, "Gaz tehlikeli seviyeye ulaştı. (Gas reached
dangerous level.)");
    String gasMessage = "Gaz seviyesi (Gas level): " + String(gasAnalog);
    telegram.send(CHAT_ID, gasMessage);
    digitalWrite(buzzerPin, HIGH);
} else if (gasAnalog < minGasAnalog) {
    telegram.send(CHAT_ID, "Gaz tehlikeli seviyeye düştü. (Gas dropped to
dangerous level.)");
    String gasMessage = "Gaz seviyesi (Gas level): " + String(gasAnalog);
    telegram.send(CHAT_ID, gasMessage);
    digitalWrite(buzzerPin, HIGH);
}
if (temperature >= minTemperature && temperature <= maxTemperature &&
pressure >= minPressure && pressure <= maxPressure && gasAnalog >=
minGasAnalog && gasAnalog <= maxGasAnalog) {
    digitalWrite(buzzerPin, LOW);
}
delay(1000);
}
```

Projeyi Gerçekleştirme Adımları:

Amaç: MQ-2 Gaz Sensörü havadaki gaz yoğunluğunu ölçerek analog veri üretir. Deneyap Basınç Ölçer modülü ise hava basıncını (hPa cinsinden) ve sıcaklığı (°C cinsinden) belirler. Tüm veriler her saniye güncellenerek seri port ekranında görüntülenir. Ölçülen değerlerin belirlenen alt ve üst eşiklerin dışına çıkması durumunda Telegram botu aracılığıyla ilgili chat ID'ye bildirim gönderilir ve buzzer aktif hale getirilerek sesli uyarı verilir.

1. Devreyi Kurma:

✓ MQ-2 Gaz Sensörü:

VCC → Deneyap Kart'taki 5V pinine

GND → Deneyap Kart'taki GND pinine

AOUT → Deneyap Kart'taki A0 pinine

✓ Aktif Buzzer:

+ → Deneyap Kart'taki D0 pinine

- → Deneyap Kart'taki GND pinine

✓ Deneyap Basınç Ölçer, I2C konnektörü ile doğrudan Deneyap Kart'a bağlanır.

4. Gerekli Yazılımlar

Projeyi çalıştırmak için bilgisayarınıza Arduino IDE'yi kurmanız gerekir:

Arduino IDE: Kod yazmak ve Deneyap Kart'a yüklemek için kullanılır.

Bilgisayarına uygun olan sürümü indir.(Windows, Mac, vs.)

Kurulumu tamamla.

Deneyap Kart 1A Tanıtımı (Board Ekleme)

Arduino IDE içinde, Deneyap Kart 1A'yı tanıtmak için aşağıdaki adımları izleyin:

1. Arduino IDE'yi açın.
2. Araçlar > Kart > Kart Yöneticisi yolunu izleyin.
3. Açılan listede "Deneyap Kart" yazın ve çıkan paketi yükleyin.

Gerekli Kütüphaneleri Yükleme:

Bu projede kullanılan sensörlerin ve Telegram özelliğinin çalışması için 2 adet kütüphane gereklidir:

1. Deneyap_BasincOlcer
2. DeneyapTelegram

Bu kütüphaneleri yüklemek için:

Arduino IDE'de üst menüden Araçlar > Kütüphane Yöneticisi yolunu izleyin.

Arama kutusuna her bir kütüphane adını yazın ve çıkan sonucu "Yükle" butonuyla yükleyin.

3. Kodu Karta Yükleme:

Kütüphaneleri yükledikten sonra verilen kodu ekrana yapıştırabilir ve aşağıdaki adımları izleyebilirsiniz.

Kodu karta yüklemek için öncelikle uygun olan kartı seçmemiz gerekmektedir.

Arduino IDE'de, Araçlar > Kart > "Deneyap Kart 1A" seçin.

Port kısmından USB'ye bağlı kartı seçin.

Araçlar > Port > (COM3, COM4, COM5 vb.)

Bu bağlantı noktası, senin bilgisayarındaki USB portuna bağlı Deneyap Kartı temsil eder.

Kart takılıysa bu kısım görünür, takılı değilse "Port" seçeneği gri olabilir.

Hangisi doğru port dersen:

Portu seçmeden önce USB'yi çıkar → listede ne var?

USB'yi tekrar tak → yeni gelen port, senin kartındır.

“✓” simgesiyle kodu derleyin.

“➡” butonuna basarak karta yükleyebilirsiniz.

4. Telegram Uyarı Sistemi

Bu projede, tehlikeli sıcaklık, gaz veya basınç durumunda:

- ✓ Buzzer çalışır.
- ✓ Telegram üzerinden mesaj gider.

Telegram mesaj örnekleri:

"Sıcaklık tehlikeli seviyeye ulaştı. (Temperature reached dangerous level.)"

"Basınç tehlikeli seviyeye ulaştı. (Pressure reached dangerous level.)"

"Gaz tehlikeli seviyeye ulaştı. (Gas reached dangerous level.)"

Hatırlatma:

Kartı doğru seçtiğinizden, kütüphaneleri doğru yüklediğinizden emin olun. Adımları gerçekleştirdiğiniz halde kod yüklenmediyse USB kablonuzun temassızlık yapıp yapmadığını kontrol edebilirsiniz.

Ek Dosya Linkleri:

Arduino IDE indirme bağlantısı: <https://www.arduino.cc/en/software/>

11. Proje Adı : Mikrofon ile Motor Kontrolü

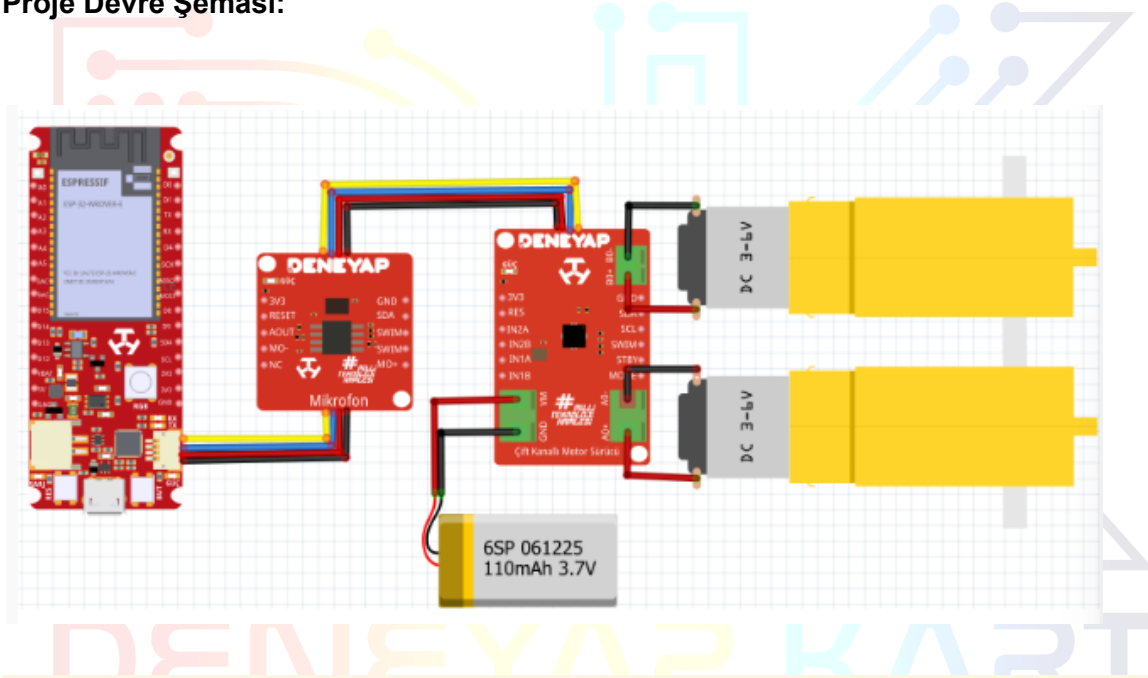
Deneyap Mikrofon ile ses ile komut alan motor kontrol uygulaması.

Deneyap Mikrofon Modülü ile tetiklenerek Deneyap Çift Kanallı Motor Sürücü ile bağlanan DC motorları kontrol eden bir sistem kurulmuştur.

Gerekli Malzemeler:

- 1 Adet Deneyap Kart 1A
- 1 Adet Li-Po Pil
- 2 Adet DC Motor
- 1 Adet Deneyap Mikrofon Sensörü
- 1 Adet Deneyap Motor Sürücü Kartı
- 4 Adet Jumper kablo
- 2 Adet I2C Bağlantı Kablosu

Proje Devre Şeması:



Proje Kodları :

```
#include <Deneyap_Mikrofon.h> // Deneyap Mikrofon kutuphanesi
eklenmesi

#include <Deneyap_CiftKanalliMotorSurucu.h> // Deneyap Çift Kanallı Motor
Sürücü kutuphanesi eklenmesi

#define threshold 520 // Verilen eşik değeri ortamın
ses durumuna göre istenilen değer ile değiştirilmeli.

DualMotorDriver MotorSurucu; // DualMotorDriver için
classtanımlanması

Microphone Mikrofon; // Microphone için class
tanımlaması

bool flag = 0;

void setup() {
    Serial.begin(115200);
```

```
if (!MotorSurucu.begin(0x16)) { // begin(slaveAddress) fonksiyonu ile
    cihazların haberleşmesi başlatılması, varsayılan:
    begin(0x16,500),frekans:500Hz
    delay(3000);
    Serial.println("I2C bağlantısı başarısız"); // I2C bağlantısı başarısız
    olursa seriport ekranına yazdırılması
    while (1);
}
if (!Mikrofon.begin(0x35)) { // begin(slaveAddress) fonksiyonu ile
    cihazların haberleşmesibaşlatılması
    delay(3000);
    Serial.println("I2C bağlantısı başarısız "); // I2C bağlantısı başarısız
    olursa serimonitore yazdırılması
    while (1);
}
pinMode(LED_BUILTIN, OUTPUT);
}
void loop() {
    uint16_t SesDeger = Mikrofon.ReadMicrophoneAnalog(); // Ses degerinin
    okunması
    Serial.print("Ses Degeri: ");
    Serial.println(SesDeger); // Ses verisi seri monitore yazıldı
    if (SesDeger > threshold) {
        flag = !flag;
        Serial.println("####Ses Algılandı####");
        Serial.print(" ");
        Serial.print(flag);
        digitalWrite(LED_BUILTIN, HIGH);
    }
    if (flag) {
        MotorSurucu.StandbyMode(ENABLE);
        MotorSurucu.MotorDrive(MOTOR1, 80, 1); // Birinci motor çalıştırılması.
        %80dutyycle hızı ile ileri yönde hareket edilmesi
        MotorSurucu.MotorDrive(MOTOR2, 80, 1); // İkinci motor çalıştırılması.
        %75dutyycle hızı ile geri yönde hareket edilmesi
        if (MotorSurucu.CheckMotorError() == 1)
            Serial.println("HATA !!! Lütfen motorları kontrol edin");
        } else
        MotorSurucu.StandbyMode(DISABLE);
        delay(100);
    }
```

Devre Kurulumu (Bağlantılar) :

1. Motor Sürücü Kartını Deneyap Kart'a I2C protokolü ile bağla.
 - o Genellikle SDA ve SCL pinleri üzerinden bağlantı kurulur.
2. DC Motorları motor sürücü çıkışlarına bağla.
 - o M1 ve M2 çıkışları motorlar için kullanılır.
3. Mikrofon Modülü yine I2C üzerinden Deneyap Kart'a bağlanır.
4. Güç Kaynağı (Pil) ile motor sürücü ve motorlara yeterli voltaj sağlanmalıdır.
 - o Deneyap Kart da USB veya pil ile beslenebilir.

Not: Bağlantılarda + ve - uçlarına dikkat etmeyi unutma. Ses sensörü ve motor sürücü modüllerinin adresleri kart üzerinden I2C ile okunur.

Adım Adım Anlatım :

1. Gerekli Malzemeleri Hazırlayın

2. Devre Bağlantılarını Yapın

a) Mikrofon Modülünü Bağlayın

- Mikrofon modülünü I2C haberleşmesiyle Deneyap Kart'a bağlayın.
- **SDA → SDA, SCL → SCL, VCC → 3.3V, GND → GND**

b) Motor Sürücü Kartını Bağlayın

- Motor sürücüyü de I2C ile Deneyap Kart'a bağlayın.
- **SDA → SDA, SCL → SCL, VCC → VIN veya 5V, GND → GND**

c) DC Motorları Bağlayın

- Motorları motor sürücü kartındaki **M1 ve M2** çıkışlarına takın.
(İstersen sadece bir motor da kullanabilirsin.)

d) Pili Bağlayın

- Pil ile motorlara ayrı güç sağlayın.
- Pili voltajı motorlara uygun olmalı (örneğin 6V).

3. Kodları Deneyap Kart'a Yükle

Arduino IDE ya da Deneyap IDE üzerinde aşağıdaki adımları izleyin:

a) Kütüphaneleri Kurun

- Deneyap_Mikrofon.h ve Deneyap_CiftKanalliMotorSurucu.h kütüphanelerini yükleyin.

b) Kodları Yükle

Kodları aşağıdaki gibi yükleyin:

4. Projeyi Çalıştırın

- Projeyi çalıştırın, seri monitörü açın.
- Ortama bir ses verin (örneğin alkış).
- Ses değeri eşik değerden yüksekse motorlar çalışmaya başlar.
- Tekrar ses verildiğinde motorlar durur.
- Yani her ses sinyali bir açma/kapama komutu gibi çalışır.

! Dikkat Edilmesi Gerekenler

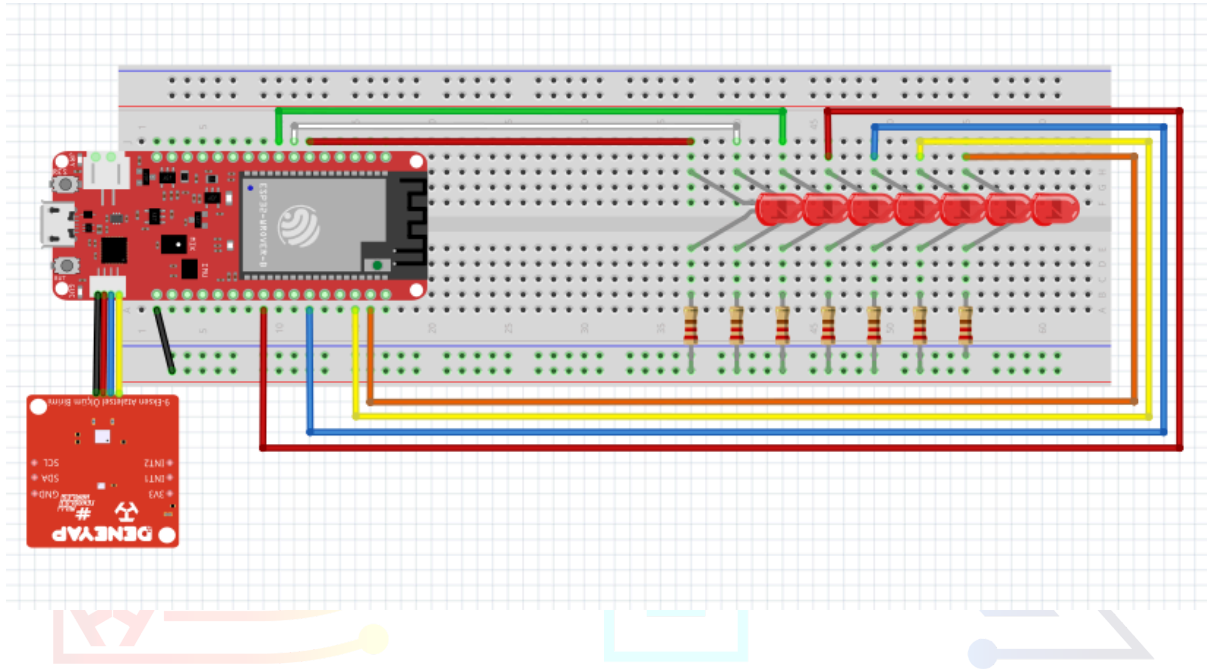
- Eşik değeri olan threshold (520), ses ortamına göre artırılıp azaltılabilir.
- Motor sürücüye bağlanan motorlar çok yüksek voltajla beslenmemeli.
- I2C bağlantılar doğru pinlere yapılmalı (SDA/SCL).

12. Proje Adı: Kart Hareketi ile LED Kontrolü

Gerekli Malzemeler:

- 1 Adet Deneyap Kart
- 1 Adet 9 Eksen Ataletsel Ölçüm Modülü
- 1 Adet Led
- 7 Adet Direnç
- 1 Adet Breadboard
- 8 Adet Jumper kablo

Proje Devre Şeması:



Proje Kodları:

```
#include <Deneyap_Eksen.h> // Zip olarak eklenmelidir:  
https://github.com/K4lender/Deneyap\_Eksen
```

```
LSM6DSM IMU;
```

```
float pitch, roll, yaw;
```

```
void setup() {
```

```
    Serial.begin(115200);
```

```
    IMU.begin(0x6B);
```

```
    pinMode(D0, OUTPUT);
```

```
    pinMode(D1, OUTPUT);
```

```
    pinMode(D12, OUTPUT);
```

```
pinMode(D13, OUTPUT);

pinMode(D4, OUTPUT);

pinMode(D8, OUTPUT);

pinMode(D9, OUTPUT);

Serial.println("Kalibrasyon başladı...");

IMU.calibrateRollPitchYaw();

Serial.println("Kalibrasyon bitti...");

}

void loop() {

    digitalWrite(D0, LOW);

    digitalWrite(D1, LOW);

    digitalWrite(D12, LOW);

    digitalWrite(D13, LOW);

    digitalWrite(D4, LOW);

    digitalWrite(D8, LOW);

    digitalWrite(D9, LOW);

    IMU.readRollPitchYaw(pitch, roll, yaw);

    Serial.print("X: ");

    Serial.println(pitch);

    Serial.print("Y: ");

    Serial.println(roll);

    Serial.print("Z: ");

    Serial.println(yaw);

    if (pitch > -5 && pitch < 5) {

        digitalWrite(D12, HIGH);

    } else if (pitch > -10 && pitch <= -5) {

        digitalWrite(D13, HIGH);

    } else if (pitch > -15 && pitch <= -10) {

        digitalWrite(D1, HIGH);

    } else if (pitch <= -15) {

        digitalWrite(D0, HIGH);

    }

}
```



```
} else if (pitch > 5 && pitch <= 10) {  
    digitalWrite(D4, HIGH);  
}  
else if (pitch > 10 && pitch <= 15) {  
    digitalWrite(D8, HIGH);  
}  
else if (pitch > 15) {  
    digitalWrite(D9, HIGH);  
}  
delay(20);  
}
```

Devre Kurulumu Adımları :

1. Breadboard'u yerleştiriyoruz.
2. Her LED'in kısa bacağı (katot) breadboard üzerindeki bir ortak eksi (GND) hattına bağlıyoruz.
3. Her LED'in uzun bacağına (anot) birer direnç bağlıyoruz.
4. Her direnç çıkışına bir jumper kablo takıp şu pinlere bağlıyoruz:
 - o D0, D1, D12, D13, D4, D8, D9
(Toplam 7 pin, her biri bir LED'i kontrol edecek.)
5. Breadboard'un eksi hattını Deneyap Kart'ın GND pinine bağlayarak devreyi tamamlıyoruz.
6. IMU sensörünü I2C bağlantısı ile Deneyap Kart'a bağlıyoruz. (Genelde SDA/SCL pinleri üzerinden)

📌 Not: IMU'nun adresi 0x6B olarak tanımlanmış, bağlantı buna göre yapılmalı.

Kodlama Adımları :

1. Kodun başında gerekli kütüphane ekleniyor:
#include <Deneyap_Eksen.h>
2. IMU başlatılıyor ve kalibrasyon yapılıyor.
- 📌 Kalibrasyon sürecinde kart sabit tutulmalı. Aksi halde eğim hesaplamaları yanlış olur.
3. setup() kısmında tüm LED pinleri çıkış olarak ayarlanıyor.
4. loop() içinde:
 - o Önce tüm LED'ler söndürülüyor (LOW).
 - o IMU üzerinden pitch (X eksen eğimi) okunuyor.
 - o Eğim değerine göre, belirlenen aralıklara karşılık gelen LED yakılıyor.

Pitch Değeri ve LED Tepkileri :

Pitch Açısı	Yakılan LED Pin
-15° ve altı	D0
-15° ile -10° arası	D1

Pitch Açısı

-10° ile -5° arası
-5° ile +5° arası
+5° ile +10° arası
+10° ile +15° arası
+15° üzeri

Yakılan LED Pin

D13
D12 (nötr)
D4
D8
D9

Ek Dosya Linkleri:

- Deneyap Kart Projeleri : <https://deneyapkart.org/projeler/kart-hareketi-ile-led-kontrolu>
- Deneyap Kart IDE indirme bağlantısı: <https://www.deneyapkart.org/>



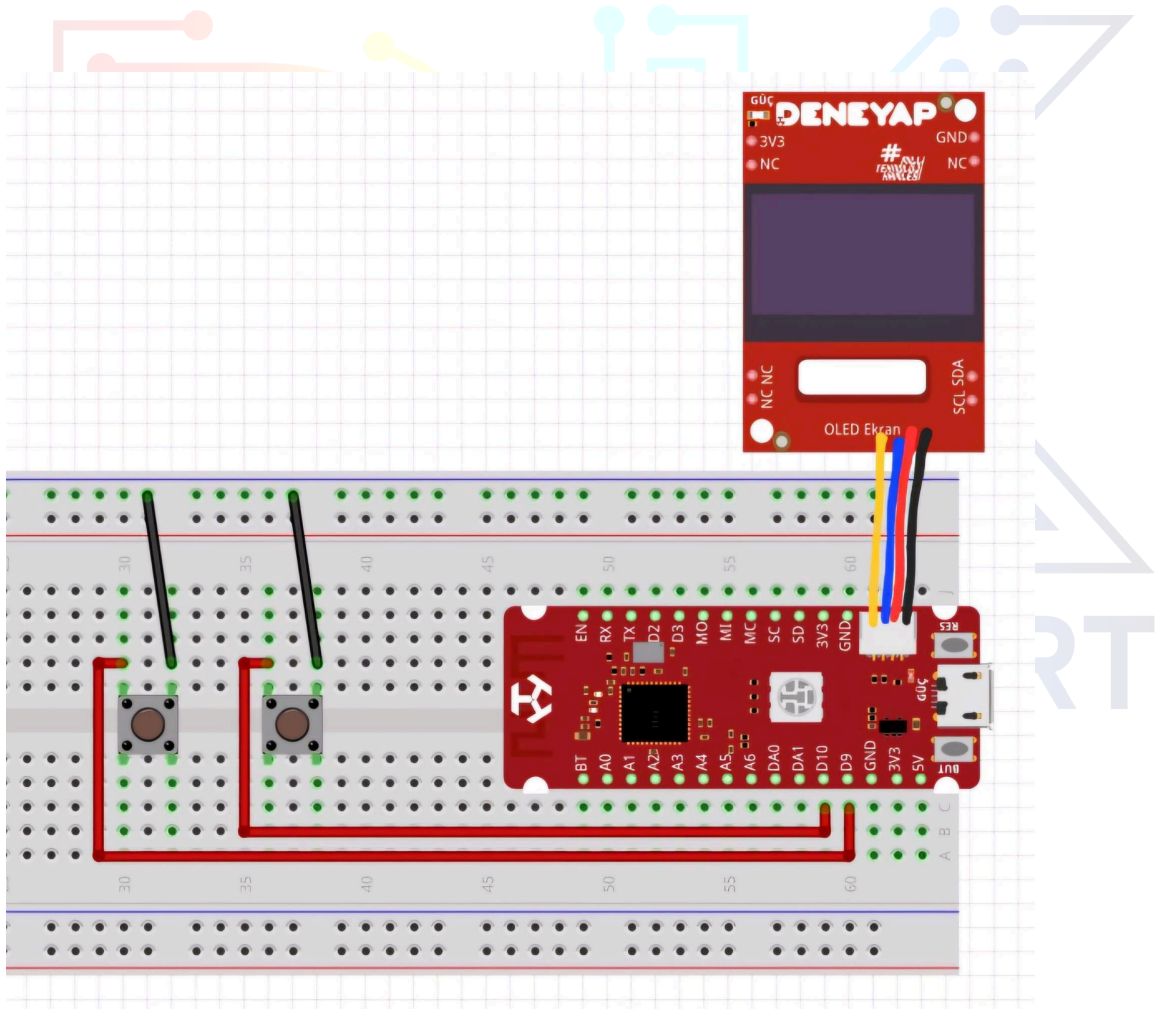
13. Proje Adı: Deneyap Mini ile XOX Oyunu

Bu projede, Deneyap Kart ve bir Deneyap Oled ekran kullanarak klasik XOX (Tic-Tac-Toe) oyununu elektronik bir formatta gerçekleştireceksiniz. XOX oyunu, kullanıcının cpu ile ızgara üzerinde X ve O sembollerini yerleştirerek oynadığı stratejik bir oyundur.

Gerekli Malzemeler:

- 1 Adet Deneyap Kart
- 1 Adet Deneyap Oled Ekran
- 2 Adet Buton
- 4 Adet Erkek- Erkek Jumper Kablo
- 1 Adet I2C kablo

Proje Devre Şeması:



Proje Kodları:

```
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>
#define BUTTON_MOVE D2
#define BUTTON_OK D3
#define OLED_RESET (-1)
Adafruit_SSD1306 display(128, 64, &Wire, OLED_RESET);
int gameStatus = 0;
int whosplaying = 0;
int winner = -1;
int board[]={0,0,0,
             0,0,0,
             0,0,0};
void playhuman() {
    int humanMove = 0;
    bool stayInLoop = true;
    bool showDot = false;
    long timerPos = millis()-1000;
    while (stayInLoop) {
        while (board[humanMove] != 0) {
            humanMove ++;
            if (humanMove >8) humanMove = 0;
        }
        if (timerPos + 200 < millis()) {
            timerPos = millis();
            showDot = !showDot;
            playhuman_showpos( humanMove, showDot);
        }
        if (digitalRead(BUTTON_MOVE)==LOW) {
            playhuman_showpos( humanMove, false);
            humanMove ++;
            while (digitalRead(BUTTON_MOVE)==LOW);
            bool showDot = false;
            long timerPos =-1000;
        }
        if (digitalRead(BUTTON_OK )==LOW) stayInLoop = false;
        delay(100);
    }
    board[humanMove] = 1;
}
void playhuman_showpos(int humanMove, bool showDot) {
    display.setTextSize(2);
    if (humanMove == 0) display.setCursor( 5, 5);
    else if (humanMove == 1) display.setCursor(25, 5);
```

```
else if (humanMove == 2) display.setCursor(45, 5);
else if (humanMove == 3) display.setCursor( 5,25);
else if (humanMove == 4) display.setCursor(25,25);
else if (humanMove == 5) display.setCursor(45,25);
else if (humanMove == 6) display.setCursor( 5,45);
else if (humanMove == 7) display.setCursor(25,45);
else if (humanMove == 8) display.setCursor(45,45);
if (showDot) display.setTextColor(WHITE); else display.setTextColor(BLACK);
display.print("?");
display.display();
}

void playcpu() {
    int cpumove = checkboard(2);
    if (cpumove >=0) {
        board[cpumove] = 2;
    }
    else {
        cpumove = checkboard(1);
        if (cpumove >=0) {
            board[cpumove] = 2;
        }
        while (cpumove < 0) {
            int randomMove = random(10);
            if (randomMove >=0 && randomMove <=8 && board[randomMove] == 0) {
                cpumove = randomMove;
            }
        }
        board[cpumove] = 2;
    }
}

int checkboard(int x){
    if (board[0]==0 && board[1]==x && board[2]==x) return 0;
    else if (board[0]==x && board[1]==0 && board[2]==x) return 1;
    else if (board[0]==x && board[1]==x && board[2]==0) return 2;
    else if (board[3]==0 && board[4]==x && board[5]==x) return 3;
    else if (board[3]==x && board[4]==0 && board[5]==x) return 4;
    else if (board[3]==x && board[4]==x && board[5]==0) return 5;
    else if (board[6]==0 && board[7]==x && board[8]==x) return 6;
    else if (board[6]==x && board[7]==0 && board[8]==x) return 7;
    else if (board[6]==x && board[7]==x && board[8]==0) return 8;
    else if (board[0]==0 && board[3]==x && board[6]==x) return 0;
    else if (board[0]==x && board[3]==0 && board[6]==x) return 3;
```

```
else if (board[0]==x && board[3]==x && board[6]==0) return 6;
else if (board[1]==0 && board[4]==x && board[7]==x) return 1;
else if (board[1]==x && board[4]==0 && board[7]==x) return 4;
else if (board[1]==x && board[4]==x && board[7]==0) return 7;
else if (board[2]==0 && board[5]==x && board[8]==x) return 2;
else if (board[2]==x && board[5]==0 && board[8]==x) return 5;
else if (board[2]==x && board[5]==x && board[8]==0) return 8;
else if (board[0]==0 && board[4]==x && board[8]==x) return 0;
else if (board[0]==x && board[4]==0 && board[8]==x) return 4;
else if (board[0]==x && board[4]==x && board[8]==0) return 8;
else if (board[2]==0 && board[4]==x && board[6]==x) return 2;
else if (board[2]==x && board[4]==0 && board[6]==x) return 4;
else if (board[2]==x && board[4]==x && board[6]==0) return 6;
else
    return -1;
}

void checkWinner() {
    winner = 3;
    if (board[0]==1 && board[1]==1 && board[2]==1) winner=1;
    else if (board[3]==1 && board[4]==1 && board[5]==1) winner=1;
    else if (board[6]==1 && board[7]==1 && board[8]==1) winner=1;
    else if (board[0]==1 && board[3]==1 && board[6]==1) winner=1;
    else if (board[1]==1 && board[4]==1 && board[7]==1) winner=1;
    else if (board[2]==1 && board[5]==1 && board[8]==1) winner=1;
    else if (board[0]==1 && board[4]==1 && board[8]==1) winner=1;
    else if (board[2]==1 && board[4]==1 && board[6]==1) winner=1;
    else if (board[0]==2 && board[1]==2 && board[2]==2) winner=2;
    else if (board[3]==2 && board[4]==2 && board[5]==2) winner=2;
    else if (board[6]==2 && board[7]==2 && board[8]==2) winner=2;
    else if (board[0]==2 && board[3]==2 && board[6]==2) winner=2;
    else if (board[1]==2 && board[4]==2 && board[7]==2) winner=2;
    else if (board[2]==2 && board[5]==2 && board[8]==2) winner=2;
    else if (board[0]==2 && board[4]==2 && board[8]==2) winner=2;
    else if (board[2]==2 && board[4]==2 && board[6]==2) winner=2;
    if (winner == 3) {
        for(int i=0;i<9;i++) if (board[i]==0) winner=0;
    }
}

void resetGame() {
    for(int i=0;i<9;i++) board[i]=0;
    winner = 0;
    gameStatus = 0;
}
```

```
void boardDrawing() {
    display.clearDisplay();
    display.setTextColor(WHITE);
    display.drawFastHLine(0, 21, 64, WHITE);
    display.drawFastHLine(0, 42, 64, WHITE);
    display.drawFastVLine(21, 0, 64, WHITE);
    display.drawFastVLine(42, 0, 64, WHITE);
    display.setTextSize(2);
    display.setCursor( 5, 5); display.print(charBoard(0));
    display.setCursor(25, 5); display.print(charBoard(1)); display.setCursor(45,
5); display.print(charBoard(2));
    display.setCursor( 5,25); display.print(charBoard(3));
    display.setCursor(25,25); display.print(charBoard(4));
    display.setCursor(45,25); display.print(charBoard(5));
    display.setCursor( 5,45); display.print(charBoard(6));
    display.setCursor(25,45); display.print(charBoard(7));
    display.setCursor(45,45); display.print(charBoard(8));
    display.display();
    delay(200);
}

String charBoard(int x) {
    if (board[x] == 0) return " ";
    if (board[x] == 1) return "o";
    if (board[x] == 2) return "x";
    return "?";
}

void setup() {
    randomSeed(analogRead(0));
    pinMode(BUTTON_MOVE,INPUT_PULLUP);
    pinMode(BUTTON_OK ,INPUT_PULLUP);
    display.begin(SSD1306_SWITCHCAPVCC, 0x3D);
    delay(500);
    display.clearDisplay();
    display.setTextColor(WHITE);
    display.display();
    whosplaying = 2;
    while ( whosplaying <0 || whosplaying > 1) whosplaying = random(2);
}

void loop() {
    if (gameStatus == 0){
        resetGame();
        boardDrawing();
    }
}
```

```
gameStatus = 1;
winner = 0;
}
if (gameStatus == 1){
    while (winner == 0) {
        display.setTextSize(2);
        if (whosplaying == 0) {
            display.setCursor( 72,25); display.print("CPU");
            display.display();
            delay(1000);
            playcpu();
            whosplaying =1;
        }
        else {
            display.setCursor( 72,25); display.print("You");
            display.display();
            playhuman();
            whosplaying =0;
        }
        boardDrawing();
        delay(500);
        checkWinner();
        if (winner > 0) {
            if (winner == 3) {
                display.setTextSize(2); display.setCursor( 68, 25);
                display.print("Draft");
            }
            else {
                display.setTextSize(2); display.setCursor( 72, 25);
                if (winner == 1) { Serial.println(F("You"));
display.print("You"); display.setCursor( 72, 45); display.print("win"); }
                else { Serial.println(F("CPU"));
display.print("CPU"); display.setCursor( 72, 45); display.print("wins");
                }
                display.display();
                delay(1000);
                while (digitalRead(BUTTON_MOVE)==HIGH && digitalRead(BUTTON_OK
)==HIGH);
            }
        }
        if (whosplaying == 0) whosplaying =1; else whosplaying =0;
        gameStatus = 0;
    }
}
```



```
delay(1000);  
}  
}  
}
```

Adım Adım Proje Anlatımı :

1. Gerekli Malzemeleri Hazırla

2. Devre Bağlantılarını Yap

OLED Bağlantısı:

- OLED ekranı I2C kablosu ile Deneyap Kart'a bağla.
- OLED genelde SDA ve SCL pinleriyle haberleşir (Deneyap Kart'ta zaten I2C portu var, oraya takman yeterli).

Buton Bağlantısı:

- Button 1 (Yön değiştirme butonu): D2 pinine bağla
- Button 2 (Seçim butonu): D3 pinine bağla
- Her butonun bir ucu GND'ye, bir ucu ilgili pine gitmeli. Pull-up direnci içsel olarak Deneyap Kart'ta ayarlanabilir, kodda o kısım var.

3. Kütüphaneleri Yükle

Arduino IDE'de şu kütüphanelerin kurulu olması gerekiyor:

- Adafruit GFX Library
- Adafruit SSD1306

Arduino IDE > Library Manager (Kütüphane Yöneticisi) > Bu isimlerle aratıp yükleyebilirsin.

4. Kodları Arduino IDE'ye Yükle

1. Arduino IDE'yi aç
2. Deneyap Kart için uygun kart tanımını yüklediğinden emin ol (Tools > Board kısmından)
3. Verdiğin tüm kodları tek bir dosyada aç
4. Kodları yüklemeyi önce kartı USB ile bağla
5. Araçlar (Tools) menüsünden portu seç (COMX gibi bir şey yazar)
6. Upload (Yükle) butonuna tıkla

Ek Dosya Linkleri:

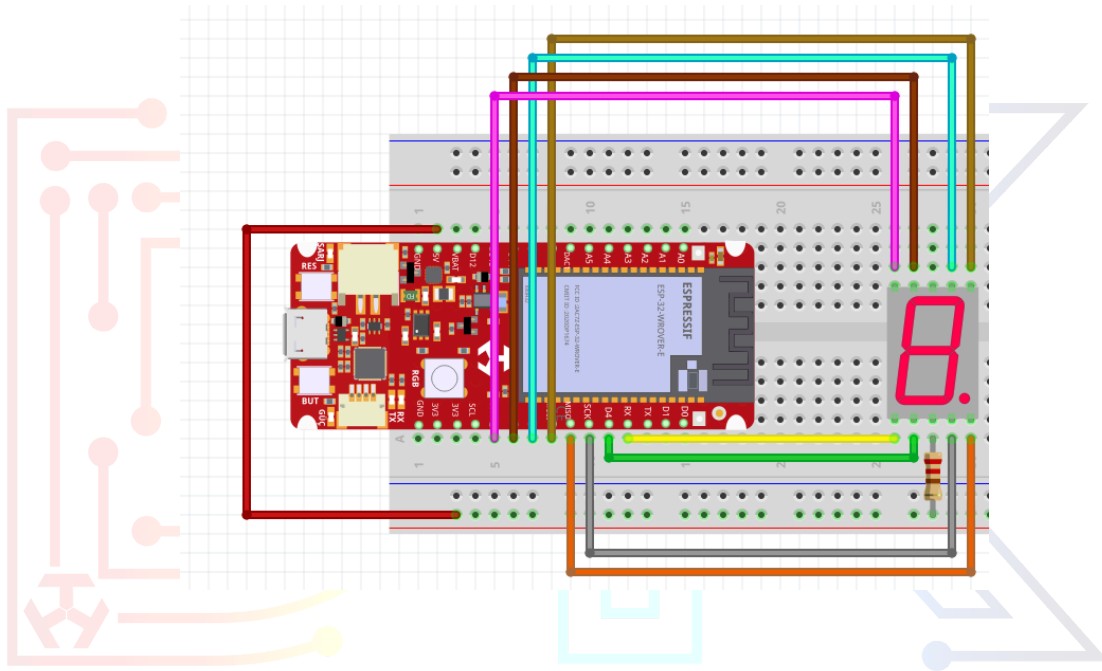
- <https://deneyapkart.org/projeler/deneyap-mini-ile-xox-oyunu>

14. Proje Adı: Deneyap Kart ile “0-9” 7 Segment Sayıcı Devre

Gerekli Malzemeler:

- Deneyap Kart
- 7 Segment Display
- Bilgisayar (Deneyap Kart IDE Kurulu)
- 220 ohm Direnç
- Jumper Kablo

Proje Devre Şeması:



Proje Kodları:

```
byte pin[ ] = {D3, D4, D5, D6, D7, D8, D9, D10}; // Deneyap Kart pin dizisi

int number[9][8] = { // Sayı dizisi
  {1, 1, 0, 0, 0, 1, 1, 1}, // 1
  {0, 0, 1, 0, 0, 0, 1, 0}, // 2
  {1, 0, 0, 0, 0, 0, 1, 0}, // 3
  {1, 1, 0, 0, 0, 1, 0, 0}, // 4
  {1, 0, 0, 0, 1, 0, 0, 0}, // 5
  {0, 0, 0, 0, 1, 0, 0, 0}, // 6
  {1, 1, 0, 0, 0, 0, 0, 1}, // 7
  {0, 0, 0, 0, 0, 0, 0, 0}, // 8
  {1, 1, 0, 0, 0, 0, 0, 0}, // 9
```

```
};  
void setup() {  
  for (byte a = 0; a < 8; a++) {  
    pinMode(pin[a], OUTPUT); // Çıkış pinlerini tanımla  
  }  
}  
  
void loop() {  
  for (int a = 0; a < 9; a++) { // 0'dan 9'a kadar olan sayılar için döngü  
    for (int b = 0; b < 8; b++) { // Her sayı için 8 segmenti kontrol et  
      digitalWrite(pin[b], number[a][b]); // Segmentleri uygun şekilde yak  
    }  
    delay(500); // 500 milisaniye bekle  
  }  
}
```

Projeyi Gerçekleştirme Adımları:

Projenin Amacı: Deneyap Kart kullanarak bir 7 segmentli ekran üzerinde 0'dan 9'a kadar sayıları sırayla görüntüleyen basit bir sayıcı devresidir. 7 segmentli ekran, her bir sayıyı göstermek için 7 LED segmentini ve bir adet nokta segmentini kullanır.

1. Devreyi Kurma:

Segment	İşlevi	Deneyap Kart Pin
a	Üst	D3
b	Üst Sağ	D4
c	Alt Sağ	D5
d	Alt	D6
e	Alt Sol	D7
f	Üst Sol	D8
g	Orta	D9
dp	Nokta	D10

GND (toprak) bacağını mutlaka breadboard'un GND hattına bağla.

Direnç (örneğin 220 ohm) bir bacağın önüne konulabilir (genellikle GND ya da ortak bacakta), bu LED'leri korur.

2. Gerekli Yazılımlar:

Projeyi çalıştırmak için bilgisayarınıza Arduino Ide'yi kurmanız gerekir:

Arduino IDE: Kod yazmak ve Deneyap Kart'a yüklemek için kullanılır.

Bilgisayarına uygun olan sürümü indir.(Windows, Mac, vs.)

Kurulumu tamamla.

Deneyap Kart 1A Tanıtımı (Board Ekleme)

Arduino IDE içinde, Deneyap Kart 1A'yı tanıtmak için aşağıdaki adımları izleyin:

1. Arduino IDE'yi açın.
2. Araçlar > Kart > Kart Yöneticisi yolunu izleyin.
3. Açılan listede "Deneyap Kart" yazın ve çıkan paketi yükleyin.

3. Kodu Karta Yükleme:

Kurulumu tamamladıktan sonra verilen kodun tamamını ekrana yapıştırabilir ve aşağıdaki adımları izleyebilirsiniz.

Kodu karta yüklemek için öncelikle uygun olan kartı ve portu seçmemiz gerekmektedir.

Arduino IDE'de, Araçlar > Kart > "Deneyap Kart 1A" seçin.

Port kısmından USB'ye bağlı kartı seçin.

Araçlar > Port > (COM3, COM4, COM5 vb.)

Bu bağlantı noktası, senin bilgisayarındaki USB portuna bağlı Deneyap Kartı temsil eder. Kart takılıysa bu kısım görünür, takılı değilse "Port" seçeneği gri olabilir.

Hangisi doğru port dersen:

- o Portu seçmeden önce USB'yi çıkar → listede ne var?
- o USB'yi tekrar tak → yeni gelen port, senin kartındır.

“✓” simgesiyle kodu derleyin.

“➡” butonuna basarak karta yükleyebilirsiniz. Kart başarılı şekilde sayıları göstermeye başlar.

4. Sonuç:

Ekranda 0'dan 9'a kadar olan sayılar sırayla yanıp söner.

Her sayı, LED segmentlerinin farklı kombinasyonlarıyla gösterilir.

Hatırlatma:

Kartı doğru seçtiğinizden, kütüphaneleri doğru yüklediğinizden emin olun. Adımları gerçekleştirdiğiniz halde kod yüklenmediyse USB kablonuzun temassızlık yapıp yapmadığını kontrol edebilirsiniz.

Ek Dosya Linkleri:

Arduino IDE İndirme bağlantısı: <https://www.arduino.cc/en/software/>

15. Proje Adı: Deneyap Kart ile Radar Yapımı

Gerekli Malzemeler:

- Deneyap Kart 1A
- Servo Motor
- HCSR04 Ultrasonik Mesafe Sensörü
- OLED Ekran
- Bilgisayar (Deneyap Kart IDE Kurulu)

Proje Devre Şeması:

Bu kısımda projeye ait fritzing çizimlerine yer verilmemiştir.

Proje Kodları:

```
#include <Deneyap_Servo.h>
#include <Deneyap_OLED.h>
#include <HCSR04.h>

#define trig D0
#define echo D1

UltraSonicDistanceSensor distanceSensor(trig, echo);
Servo servo;
OLED oled;

int mesafe;
uint16_t pos;

void setup() {
    // put your setup code here, to run once:
    Serial.begin(115200);
    servo.attach(D4);
    oled.begin(0x7A);
    oled.clearDisplay();
    oled.setTextXY(0, 0);
    oled.putString("*Deneyap Radar*");
    oled.setTextXY(7, 0);
    oled.putString("*****");
}

void loop() {
    // put your main code here, to run repeatedly:
    for (pos = 0; pos < 180; ++pos) {
```

```
servo.write(pos);
mesafe = distanceSensor.measureDistanceCm();
Serial.print("Mesafe = ");
Serial.println(mesafe);
if (pos % 2 == 0) {
    oled.setTextXY(3, 0);
    oled.putString("Mesafe = ");
    oled.setTextXY(3, 9);
    oled.putFloat(mesafe);
}
delay(50);
}

for (pos = 180; pos > 0; --pos) {
    servo.write(pos);
    mesafe = distanceSensor.measureDistanceCm();
    Serial.print("Mesafe = ");
    Serial.println(mesafe);
    if (pos % 2 == 0) {
        oled.setTextXY(3, 0);
        oled.putString("Mesafe = ");
        oled.setTextXY(3, 9);
        oled.putFloat(mesafe);
    }
    delay(50);
}
}
```

Projeyi Gerçekleştirme Adımları:

1. Devreyi Kurma:

- ♦ Ultrasonik Sensör (HC-SR04):

Sensör Pin	Deneyap Kart Pin
VCC	5V
GND	GND
TRIG	D0
ECHO	D1

- ♦ Servo Motor (SG90):

Motor Kablosu	Deneyap Kart Pin
Kırmızı (VCC)	5V
Kahverengi (GND)	GND
Turuncu (Sinyal)	D4

- ♦ OLED Ekran (I2C - SSD1306):

OLED Pin	Deneyap Kart Pin
VCC	3.3V
GND	GND
SDA	D2
SCL	D3

- ✓ Oled Ekran, I2C konnektörü ile doğrudan Deneyap Kart'a bağlanır.

2. Gerekli Yazılımlar

Projeyi çalıştırmak için bilgisayarınıza Arduino Ide'yi kurmanız gerekir:

Arduino IDE: Kod yazmak ve Deneyap Kart'a yüklemek için kullanılır.

Bilgisayarına uygun olan sürümü indir.(Windows, Mac, vs.)
Kurulumu tamamla.

Deneyap Kart 1A Tanıtımı (Board Ekleme)

Arduino IDE içinde, Deneyap Kart 1A'yı tanıtmak için aşağıdaki adımları izleyin:

1. Arduino IDE'yi açın.
2. Araçlar > Kart > Kart Yöneticisi yolunu izleyin.
3. Açılan listede "Deneyap Kart" yazın ve çıkan paketi yükleyin.

Gerekli Kütüphaneleri Yükleme:

Üstte "Araçlar > Kütüphane Yöneticisi" seç.

Şunları sırayla ara ve yükle:

- Deneyap_Servo
- Deneyap_OLED
- HCSR04

Bu kütüphaneler, sensörleri ve motoru tanıyabilmek için gereklidir.

Arama kutusuna her bir kütüphane adını yazın ve çıkan sonucu **"Yükle"** butonuyla yükleyin.

3. Kodu Karta Yükleme:

Kütüphaneleri yükledikten sonra verilen kodun tamamını ekrana yapıştırabilir ve aşağıdaki adımları izleyebilirsiniz.

Kodu karta yüklemek için öncelikle uygun olan kartı ve portu seçmemiz gerekmektedir.

Arduino IDE'de, Araçlar > Kart > "Deneyap Kart 1A" seçin.

Port kısmından USB'ye bağlı kartı seçin.

Araçlar > Port > (COM3, COM4, COM5 vb.)

Bu bağlantı noktası, senin bilgisayarındaki USB portuna bağlı Deneyap Kartı temsil eder. Kart takılıysa bu kısım görünür, takılı değilse "Port" seçeneği gri olabilir.

Hangisi doğru port dersen:

Portu seçmeden önce USB'yi çıkar → listede ne var?

USB'yi tekrar tak → yeni gelen port, senin kartındır.

“✓” simgesiyle kodu derleyin.

“➡” butonuna basarak karta yükleyebilirsiniz.

4.Nasıl Çalışır?

- o Servo motor sağa sola döner.
- o Üzerindeki sensör her açıdan mesafeyi ölçer.
- o Ölçülen mesafe hem seri monitörde hem OLED ekranda görünür.

5.Sonuç:

Sensör çalışıyor mu diye bak:

- Arduino IDE’de Araçlar > Seri Monitör aç → Mesafeler görünmeli.
- OLED ekranda da "Mesafe = xx cm" yazar.

Hatırlatma:

Kartı doğru seçtiğinizden, kütüphaneleri doğru yüklediğinizden emin olun. Adımları gerçekleştirdiğiniz halde kod yüklenmediyse USB kablonuzun temassızlık yapıp yapmadığını kontrol edebilirsiniz. OLED ekran bozuksa, 0x7A adresini 0x3C yapmayı deneyebilirsiniz. Eğer servo titriyorsa, harici güçle beslemeyi düşünebilirsiniz.

Ek Dosya Linkleri:

Arduino IDE indirme bağlantısı: <https://www.arduino.cc/en/software/>

Proje Detayları: <https://deneyapkart.org/projeler/deneyap-kart-ile-radar-yapimi-2>

DENEYAP KART


```
/*
 * Çift Kanallı DC Motor ile çizgi izleyen robot örneği,
 * Bu algılayıcı I2C haberleşme protokolü ile çalışmaktadır.
 *
 * ----->
https://docs.deneyapkart.org/tr/content/contentDetail/deneyap-module-deneyap-dual-channel-motor-driver-m <-----
 *
 * -----> https://github.com/DeneyapKart/deneyap-cift-kanalli-motor-surucu-arduino-library <-----
 */

#include <Deneyap_CiftKanalliMotorSurucu.h> // Deneyap Çift Kanallı Motor Sürücü kütüphanesi eklenmesi

DualMotorDriver MotorDriver; // DualMotorDriver için sınıf tanımlanması
// Çizgi izleyen robotun dijital pinlerin tanımlanması
```

```
const int digitalPin0 = D7;
const int digitalPin1 = D6;
const int digitalPin2 = D5;

void setup() {
    Serial.begin(115200); // Seri haberleşmenin başlatılması
    // I2C bağlantısının başlatılması ve kontrol edilmesi
    if (!MotorDriver.begin(0x16)) { // begin(slaveAdress)
        // fonksiyonu ile cihazların haberleşmesi başlatılması
        delay(3000);
        Serial.println("I2C bağlantısı başarısız"); // I2C bağlantısı başarısız
        // olursa seri port ekranına yazdırılması
        while (1); // Bağlantı başarısızsa sonsuz döngü
    }
}

void loop() {
    // Dijital pinlerin okunması
    int digitalValue0 = digitalRead(digitalPin0);
    int digitalValue1 = digitalRead(digitalPin1);
    int digitalValue2 = digitalRead(digitalPin2);
    // Motor hatasının kontrol edilmesi
    if (MotorDriver.CheckMotorError() == 1) {
        Serial.println("HATA !!! Lütfen motorları kontrol edin");
    }
    // Motor sürücü kontrolü
    motorControl(digitalValue0, digitalValue1, digitalValue2);
}
// Motor sürücü kontrol fonksiyonu
void motorControl(int value0, int value1, int value2) {
    if (value2 == 1) {
        MotorDriver.MotorDrive(MOTOR1, 0, 1);
        MotorDriver.MotorDrive(MOTOR2, 45, 0); // 2. motor %45 dutycycle ile
        // ileri yönde çalıştırılması
    } else if (value1 == 1 && value0 == 0 && value2 == 0) {
        MotorDriver.MotorDrive(MOTOR1, 35, 1); // 1. motor %35 dutycycle ile geri
        // yönde çalıştırılması
        MotorDriver.MotorDrive(MOTOR2, 35, 0); // 2. motor %35 dutycycle ile
        // ileri yönde çalıştırılması

        delay(2);
    } else if (value1 == 0 && value0 == 0 && value2 == 0) {
```

```
MotorDriver.MotorDrive(MOTOR1, 30, 1); // 1. motor %30 dutycycle ile geri  
yönde çalıştırılması  
MotorDriver.MotorDrive(MOTOR2, 30, 0); // 2. motor %30 dutycycle ile  
ileri yönde çalıştırılması  
delay(2);  
} else if (value0 == 1) {  
    MotorDriver.MotorDrive(MOTOR1, 45, 1); // 1. motor %45 dutycycle ile geri  
yönde çalıştırılması  
    MotorDriver.MotorDrive(MOTOR2, 0, 0);  
}  
}
```

Yapım Aşaması:

1. Malzemelerin temini

2. Mekanik Tasarım:

- Şase üzerine motorlar yerleştirilip sabitlenmelidir.
- Tekerlekler motor millerine takılmalı, ön kısma ise denge tekeri yerleştirilmelidir.
- Deneyap Kart, şase üzerine sabitlenmeli ve çizgi sensörleri yere bakacak şekilde aracın ön tarafına yerleştirilmelidir.
- Deneyap Çift Kanallı Motor Sürücü, Deneyap Kart ile uygun I2C adresi üzerinden haberleşecek şekilde bağlanmalıdır.
- Piller de aracın dengesini bozmayacak bir şekilde araç üstündeki boş kısımlara yerleştirilmelidir.

3. Elektronik Tasarım:

- Bağlantılar devre şemasına uygun bir biçimde yapılmalıdır.

4. Yazılım Tasarım:

- Kullanılan malzemeler için gerekli kütüphaneler indirilmelidir.
- Yukarıda yer alan kodlar kullanılabilir.
- Kodlarda yapılan işlemler sırasıyla açıklanmıştır, anlaşılmayan kısımlar ve olası sorunlar oradan tespit edilebilir.

Hatırlatma:

- **Güç Yönetimi:** Motorlar çalışırken voltaj düşüşü yaşanabilir. Güç kaynağının yeterli olduğundan emin olun.
- **Tekerlek hizalaması:** Şase üzerindeki motorların dengeli yerleştirilmiş olması gerekir, aksi halde robot düz gitmez.
- **Pin Tanımlamaları:** Pin tanımlamaları örnek devreye uygun biçimde yapılmıştır. Olası değişikliklerde bunları kodlarda düzenlemeyi unutmayınız.

Ek Dosya Linkleri:

<https://deneyapkart.org/projeler/deneyap-kart-ile-cizgi-izleyen-robot>