



SAKARYA
ÜNİVERSİTESİ

NESNELERİN İNTERNETİ VE UYGULAMALARI PROJE ÖDEVİ

Emre Özel

B191210086

1. ÖĞRETİM A GRUBU

İÇİNDEKİLER

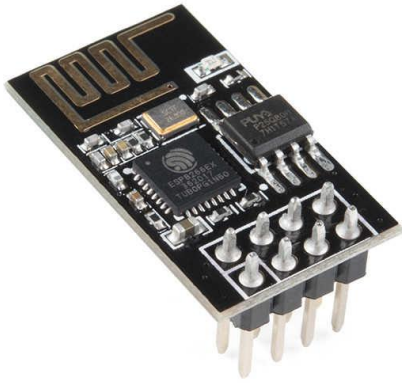
- Giriş
- Projede Kullanılan Gereçler
- Business Canvas İş Modeli
- Big Data(Büyük Veri)
- Proje kodu
- Kaynakça

GİRİŞ

Bu projede araç park sensörü yaptım . Projemde esp8266 nodemcu , buzzer,led, hc sr04 ultrasonik mesafe sensörü ve breadboard kullandım. Amacım projemin kullanıldığı benzin deposunda benzin miktarı azaldıkça artan uyarı sesleri oluşturmak.

PROJEDE KULLANILAN GEREÇLER,

NodeMcu LoLin ESP8266



NodeMCU üzerinde ESP8266 modülü bulunduran açık kaynak kodlu, ufak boyutlu elektronik geliştirme kartıdır. Ucuz olmasına rağmen çok stabil çalışan bir karttır. Kullanım alanı oldukça geniştir. Üzerinde bulunan **ESP8266** Wifi modülü sayesinde internete kolay bir şekilde bağlanabiliyor, bu özelliği sayesinde uzaktan kontrol ve **IOT** projelerinde çok fazla kullanılır. Ayrıca düşük güç tükettiği için, güç tüketimi önemli olan projelerde de çok tercih edilir.

Buzzer



Buzzer, verilen voltaja göre farklı ses sinyalleri sağlayan bir cihazdır. Maliyetleri az, üretimi basit ve çok hafif yapıda olmalarından dolayı kullanım alanı çok geniştir.

HC-SR04 Ultrasonik mesafe sensörü



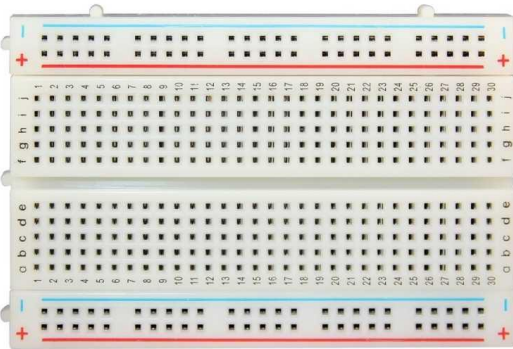
HC- SR04 Ultrasonik Sensör sonar iletişim kullanarak karşısındaki nesneye olan mesafeyi hesaplayan bir kaynaktır. Sonar dediğimiz sistem ses dalgalarını kullanarak cismin uzaklığı hesaplamamıza yardımcı olur. Bu tür **sensörlerin** esin kaynağı yunuslar ve yarasalardır.

ARDUINO IDE



Arduino IDE, arduino kitleri için geliřtirdiđi; komutların yazılmasına, derleme iřleminin yapılmasına ve son olarakta derlenen kodları dođrudan (Bilgisayarın USB portuna bađlı olan) Arduino kite yüklenmesine olanak sađlayan yazılım geliřtirme platformudur.

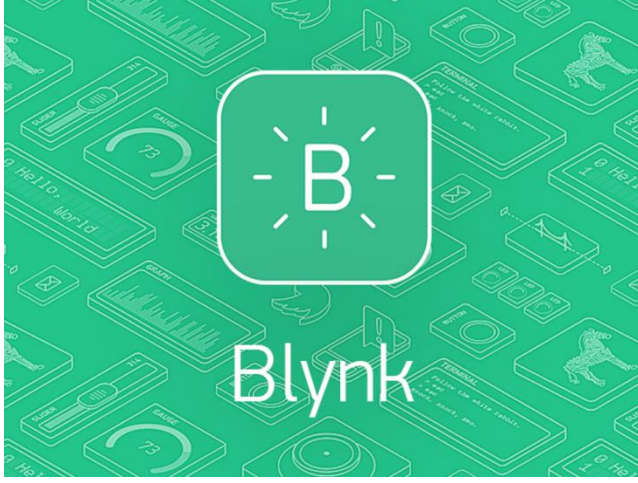
BREADBOARD



Breadboard üzerinde devrelerimizi test ettiđimiz araçtır. Kurduđumuz devreleri birbirlerine lehimlemeden kolaylıkla test etmemizi sađlar. Tasarladıđımız devreleri baskı devre veya delikli plaketler üzerine aktarmadan önce denememize olanak sađlar. Bu sayede devre bađlantılarını kontrol

ederek bir hata olup olmadığını gözlemlemiş oluruz. Devreleri tak-çıkâr şeklinde kurabildiğimiz için kullandığımız elektronik bileşenleri başka projelerde tekrar kullanma imkanı verir.

BLYNK



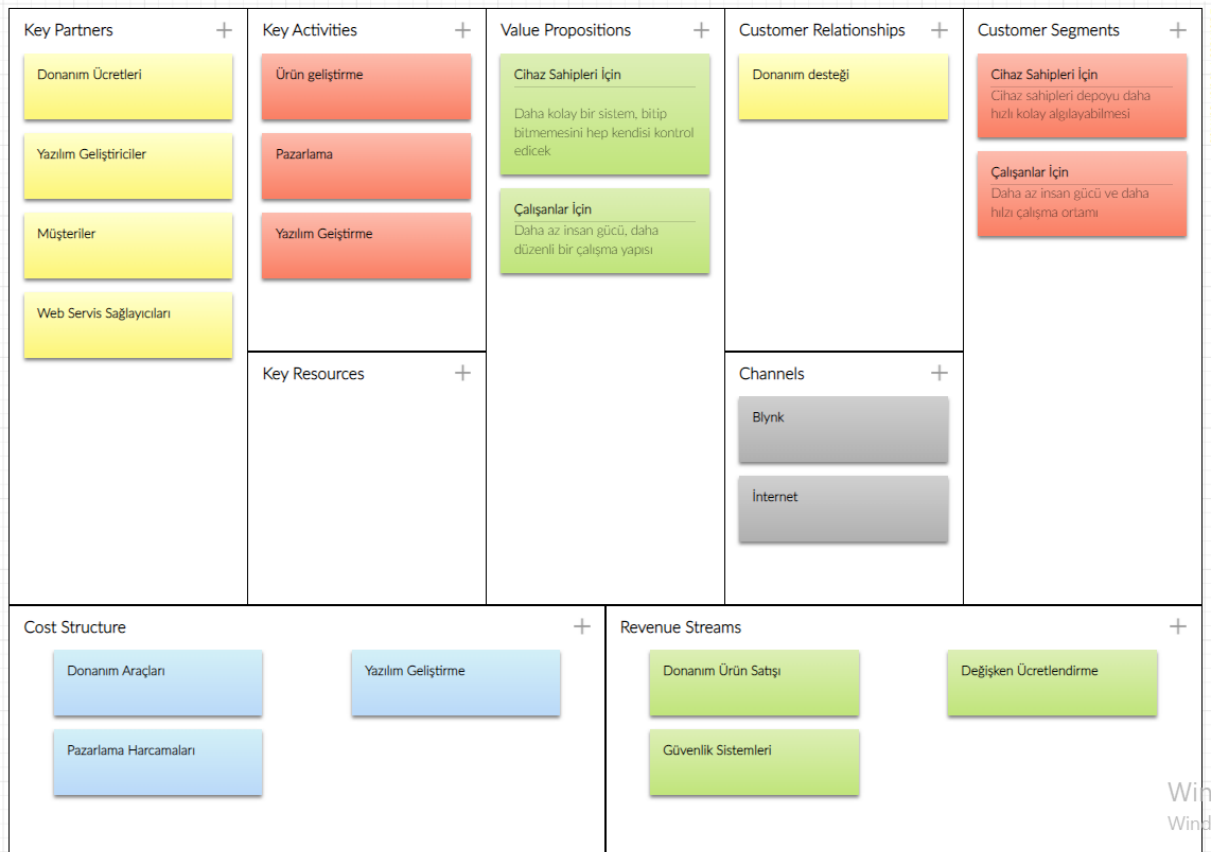
Blynk ile Arduino, Raspberry Pİ gibi benzer kartları kontrol etmenize olanak sağlayan bir platformdur.

Blynk nesnelerin interneti için tasarlanmıştır. Bağlı donanımları uzaktan kontrol edebilir, sensör verilerini okuyabilir, verileri görselleştirebilir, depolayabilir ve bir çok daha güzel şeyler yapabilir.



İngilizce açılımı Light Emitting Diode olan **LED**, ışık yakan diyot olarak tanımlanmaktadır. **LED**, yarı iletken bir devre elemanıdır. **LED**, içerisinde bir elektrik akımı geçtiğinde ışık yayan bir sisteme sahiptir. Işık, yarı iletken malzeme içerisinde akımı taşıyan parçacıklarla buluştuğunda üretilir.

BUSINESS CANVAS MODEL



BIG DATA(BÜYÜK VERİ)

Birbirine bağlı çok sayıda fiziksel cihazın veya sensörün internet üzerinden ürettiği veri büyük veri (big data) olarak adlandırılır. Büyük veri, kavramsal olarak verinin hacmini ifade etmektedir. Büyük veri, büyük hacimli, yüksek hız ve çok çeşitli verilerin yönetimi ve analizini ifade etmektedir.

Sürekli kullanılan sosyal medya hesaplarındaki etkileşimler, arama motorları ve buralarda arama yapıldığında arkada bırakılan izler, banka hesaplarıyla yapılan hareketler, bloglar, mailler, sensörler ve tekil kullanıcıların internet ile olan tüm etkileşimleri bir araya getirilince büyük bir veri yığını oluşmuş oluyor. Big datayı, bu yığından beslenip anlamlı ve işlenebilir bilgileri kullanıcıya sunan bir hazine olarak değerlendirebiliriz.

IOT Verileri saklanmalı, işlenebilmeli ve anlamlandırılmalıdır. Saklama için verimli ve efektif disk alanına ihtiyaç vardır. Big Data kabul edilebilir bir sürede işlenebilir olmalıdır. Bulut sistemler uzaktan yönetilebilir, istenildiği zaman boyutu ölçeklenebilir yapıdadır.

PROJE KOD

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

#define echoPin D6
#define trigPin D7
#define buzzerPin D5
#define led D8
int maximumRange = 50;
```



```

int minimumRange = 0;
int olcum=0;

BlynkTimer timer;
char auth[] = "9o6KesXz960UvU6qN5WFsvCRY_f-TXsD"; //Auth code sent via
Email
char ssid[] = "MyPhone";
char pass[] = "Emre.2648";

void melodi(int dly)
{
    tone(buzzerPin, 440);
    delay(dly);
    noTone(buzzerPin);
    delay(dly);
}
void notifyOnFire()
{
    olcum = mesafe(maximumRange, minimumRange);
    Serial.println(olcum);

    //Blynk.notify(olcum);
}
void setup()
{
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    pinMode(buzzerPin, OUTPUT);
    pinMode(led, OUTPUT);
    Serial.begin(115200);
    Blynk.begin(auth, ssid, pass);
    timer.setInterval(1000L, notifyOnFire);
}
void loop()
{
    Blynk.run();
    timer.run();
}
int mesafe(int maxrange, int minrange)
{
    long duration, distance;

    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);

    duration = pulseIn(echoPin, HIGH);
    distance = duration / 58.2;
    //delay(50);

    if(distance >= maxrange || distance <= minrange)
    {
        Blynk.notify("Dikkat");
        digitalWrite(led, HIGH);
    }
}

```

```
        tone(buzzerPin, 440);

    return 0;
}
int gurultu = (50-olcum)*10;
melodi(gurultu);
digitalWrite(led,LOW);
return distance;
}
```

KAYNAKÇA

- <https://tr.wikipedia.org/>