T.C

YÜKSEK İHTİSAS ÜNİVERSİTESİ

MESLEK YÜKSEK OKULU



# RİNZLER

2005010019 Mustafa ÖZDEMİR

2005010004 Şule Melda ÖZÇAKIR

Bilgisayar Programcılığı

Mayıs 2022

İÇİNDEKİLER

1. Projenin Amacı
2. Giriş

* Arduino nedir?
* Arduino ile neler yapılabilir?
* Arduino Uno R3 nedir?
* Ne işe yarar? Nasıl kullanılır?

# Metot (Yöntem)

1. Projenin Gerçekleştirilmesi ve Proje Görselleri
2. Sonuç ve Tartışma
3. Kaynakça

**PROJENİN AMACI**

Arduino tabanlı olarak geliştirdiğimiz projemizde ki amacımız Mikroişlemci ve Assembler dersinde öğrendiğimiz bilgileri temel alarak elektronik bir kart yapmak ve bu kartla bağlantı kurduğumuz bilgisayar klavyelerinin kodlamasını bozarak kendi kodlarımızın bu klavyede çalışmasını sağlamak.

Bu projemizin alt yapısını oluştururken https://lezzetlirobottarifleri.com/ sitesinde hazırlanmış olan bir projeyi temel alıp üzerine kendi kodlarımızı ve kendi kart düzeneğimizi ekleyerek yeni bir proje oluşturduk.

Burada ki asıl amaç Arduino IDE kullanarak pratik klavye simülasyonunu gerçekleştirerek klavyemizde kendi kumanda konrolümüzü yapmış olduk.

**GİRİŞ**

**Arduino Nedir?**

**Arduino**, interaktif projeler geliştirmek için tasarlanan, elektronik donanım ve yazılım temelli bir geliştirme platformudur.

Arduino kartlarında bir adet **Atmel AVR mikrodenetleyici** ve devre bağlantıları için çeşitli elektronik komponentler bulunur.

Arduino, İtalyan mühendisler tarafından geliştirilmiştir.

Tamamen açık kaynak kodlu bir platformdur.

**Arduino ile Neler Yapılabilir?**

Arduino’ya bağlanan sensörler vasıtasıyla ortam etkileşimli giriş sinyalleri elde edip bu sinyalleri incelemek ve bu sinyaller ile algoritmalar kurarak çıkış sinyalleri üretmek mümkündür. Bu sayede çeşitli robotlar ve elektronik sistemler tasarlanabilir.

Kısacası Arduino ile robotik, drone, otomasyon sistemleri, akıllı sistemler, uzaktan kumandalı araçlar, giyilebilir teknoloji projeleri ve sayısız birçok proje yapılabilir.

**Arduino Uno R3 Nedir?**

Arduino uno r3, kendine has ve kolay Arduino kodlama ([Arduino IDE](https://www.arduino.cc/en/main/software" \t "_blank)) diliyle programlanabilen geliştirme kartı olarak üretilmiştir. Arduino, dahili programlamayı geleneksel yöntemlerden daha kolay hale getiren mikrodenetleyici barındıran bir platformdur.

****

**Ne İşe Yarar? Nasıl Kullanılır?**

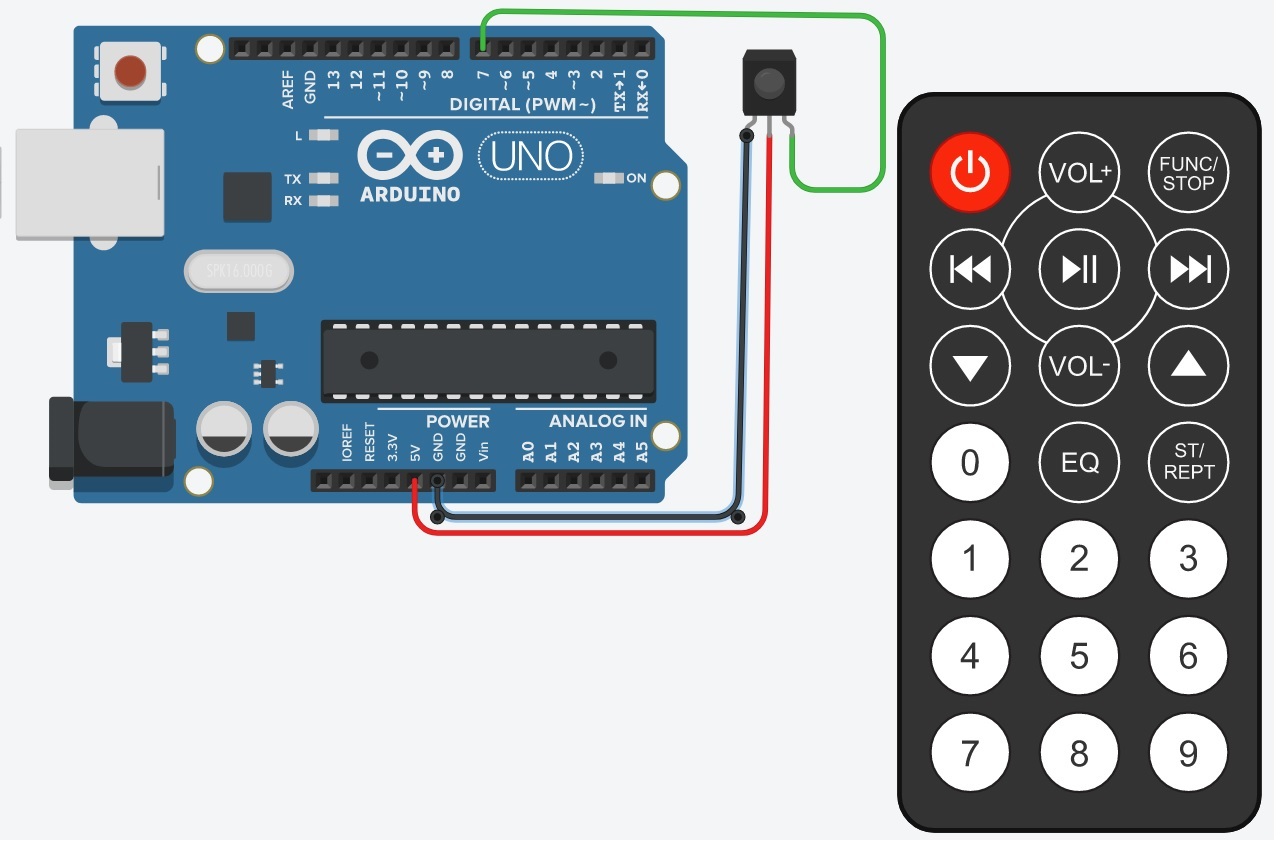
Arduino çeşitleri, bilgisayaranıza USB yoluyla bağlayıp [Arduino IDE](https://www.arduino.cc/en/Main/Software" \t "_blank) arayüzünü kullanarak dilediğiniz şekilde kodlama yapabileceğiniz bir geliştirme kartıdır.

Üstelik bunun için programlama dillerini bilmenize de gerek yok. İnternet üzerinde hali hazırda açık kaynak olarak kod paylaşımı yapan (yani kopyala yapıştır yapabileceğiniz) bir çok site var.



**METOT (YÖNTEM)**

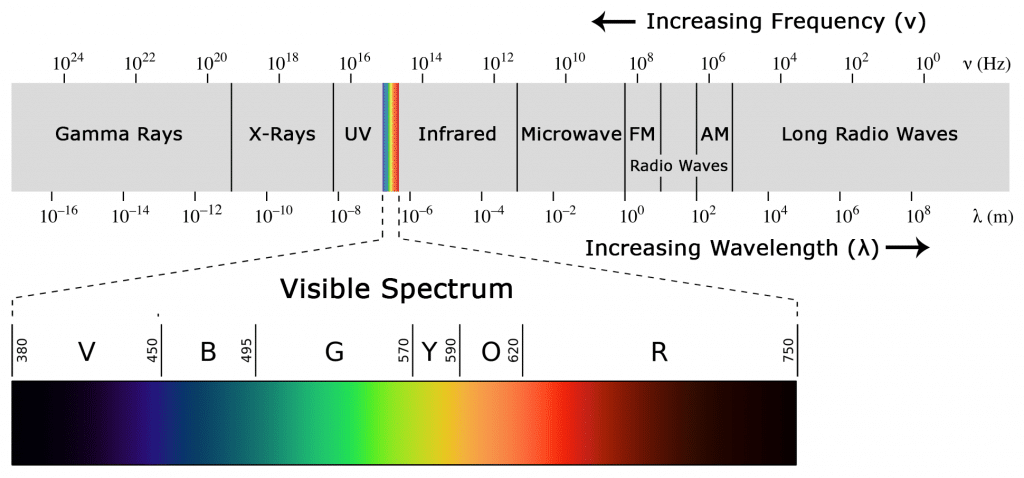
Devre Şeması ve Kullanılan Malzemeler



* IR receiver
* Arduino Uno R3
* Jumper Cable 20cm (3 adet)
* Remote Control

**KIZILÖTESİ NEDİR?**

Kızılötesi radyasyon, etrafımızda gördüğümüz ışığa benzer bir ışık şeklidir. IR ışığı ile görünür ışık arasındaki tek fark, frekans ve dalga boyudur. Kızılötesi radyasyon, görünür ışık aralığının dışındadır, bu nedenle insanlar onu göremez:

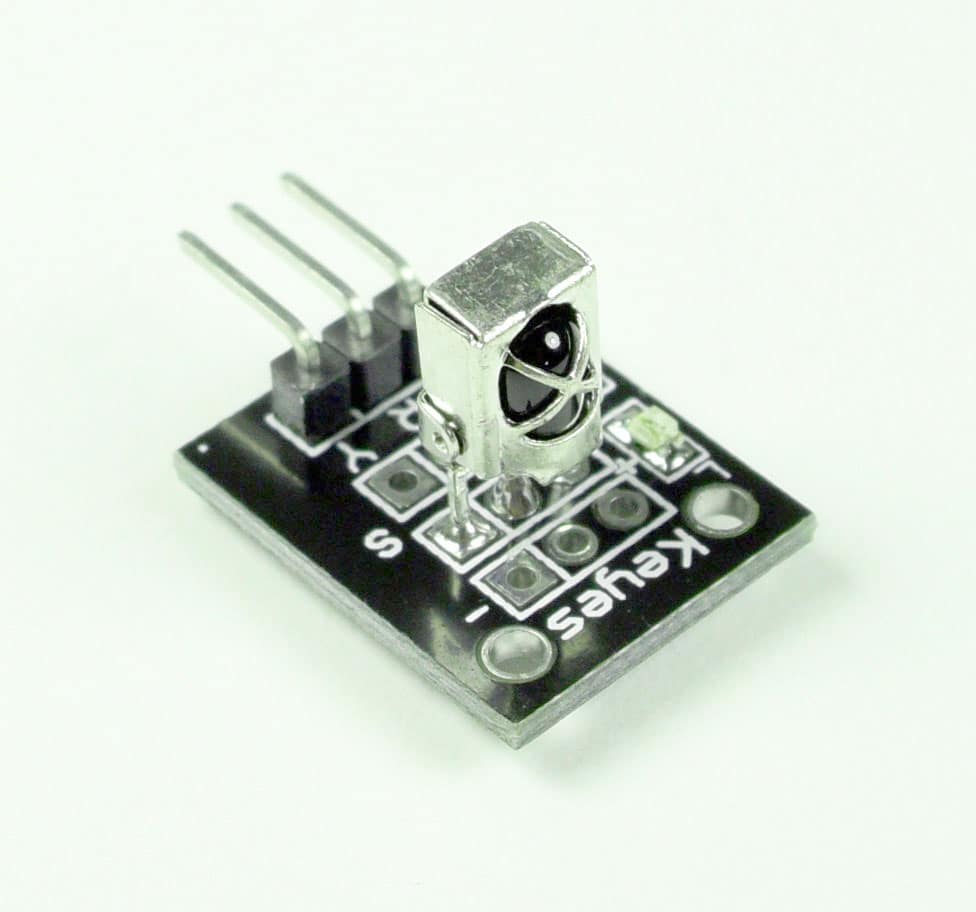
[](file:///C:\Users\User\Desktop\)

IR bir tür ışık olduğu için, IR iletişimi alıcıdan vericiye doğrudan bir görüş hattı gerektirir. Duvarlardan veya WiFi veya Bluetooth gibi diğer materyallerden iletemez.

**IR UZAKTAN KUMANDALAR VE ALICILAR NASIL ÇALIŞIR?**

Tipik bir kızılötesi iletişim sistemi, bir IR vericisi ve bir IR alıcısı gerektirir. Verici, görünür spektrum yerine IR spektrumunda ışık üretmesi dışında standart bir LED'e benziyor. Bir TV uzaktan kumandasının önüne bakarsanız, IR verici LED'ini görürsünüz:

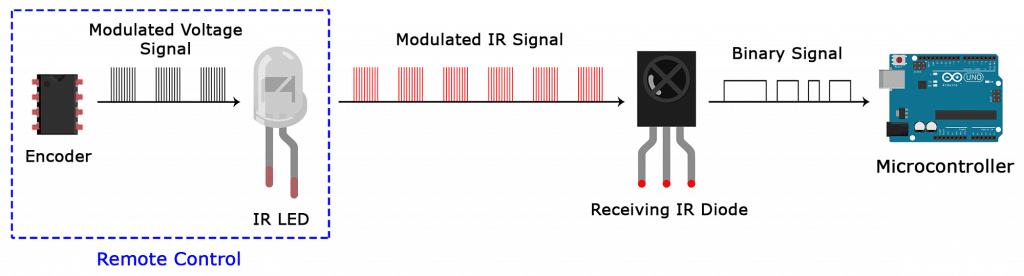
[](file:///C:\Users\User\Desktop\)

[](file:///C:\Users\User\Desktop\)

### IR SİNYAL MODÜLASYONU

IR ışığı güneş, ampuller ve ısı üreten diğer her şey tarafından yayılır. Bu, etrafımızda çok fazla IR ışık gürültüsü olduğu anlamına gelir. Bu gürültünün IR sinyaline karışmasını önlemek için bir sinyal modülasyon tekniği kullanılır.

IR sinyal modülasyonunda, IR uzaktan kumanda üzerindeki bir kodlayıcı, ikili sinyali modüle edilmiş bir elektrik sinyaline dönüştürür. Bu elektrik sinyali verici LED'e gönderilir. Verici LED, modüle edilmiş elektrik sinyalini modüle edilmiş bir IR ışık sinyaline dönüştürür. IR alıcısı daha sonra IR ışık sinyalini demodüle eder ve bilgiyi bir mikro denetleyiciye aktarmadan önce onu tekrar ikiliye dönüştürür:

[](file:///C:\Users\User\Desktop\)

Modüle edilmiş IR sinyali, taşıyıcı frekansı olarak bilinen yüksek bir frekansta açılıp kapatılan bir dizi IR ışık darbesidir. Çoğu verici tarafından kullanılan taşıyıcı frekansı 38 kHz'dir, çünkü doğada nadirdir ve bu nedenle ortam gürültüsünden ayırt edilebilir. Bu şekilde IR alıcısı, 38 kHz sinyalinin vericiden gönderildiğini ve çevredeki ortamdan alınmadığını bilecektir.

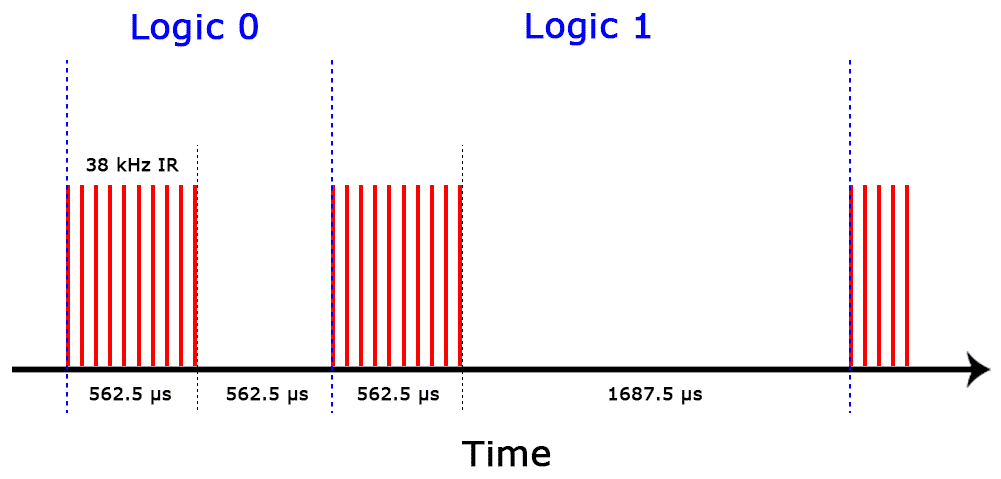
Alıcı diyot, IR ışığının tüm frekanslarını algılar, ancak bir bant geçiren filtreye sahiptir ve yalnızca 38 kHz'de IR'ye izin verir. Daha sonra modüle edilmiş sinyali bir ön yükseltici ile yükseltir ve bir mikro denetleyiciye göndermeden önce onu ikili bir sinyale dönüştürür.

### IR İLETİM PROTOKOLLERİ

Modüle edilmiş IR sinyalinin ikiliye dönüştürüldüğü model, bir iletim protokolü ile tanımlanır. Birçok IR iletim protokolü vardır. Sony, Matsushita, NEC ve RC5 daha yaygın protokollerden bazılarıdır.

NEC protokolü aynı zamanda Arduino projelerinde en yaygın türdür, bu yüzden alıcının modüle edilmiş IR sinyalini ikiliye nasıl dönüştürdüğünü size göstermek için bir örnek olarak kullanacağım.

Mantıksal '1', 38 kHz IR'lik 562,5 µs uzunluğunda bir **YÜKSEK** darbe ile başlar ve bunu 1,687,5 µs uzunluğunda bir **DÜŞÜK** darbe izler. Mantıksal '0', 562,5 µs uzunluğunda bir YÜKSEK darbe ve ardından 562,5 µs uzunluğunda bir DÜŞÜK darbe ile iletilir:

[](file:///C:\Users\User\Desktop\)

NEC protokolü, ikili verileri modüle edilmiş bir sinyale bu şekilde kodlar ve çözer. Diğer protokoller, yalnızca bireysel YÜKSEK ve DÜŞÜK darbelerin süresinde farklılık gösterir.

### IR KODLARI

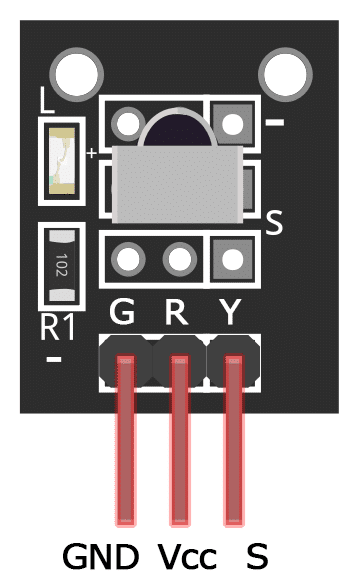
Uzaktan kumandadaki bir düğmeye her bastığınızda, benzersiz bir onaltılık kod oluşturulur. Modüle edilen ve IR üzerinden alıcıya gönderilen bilgidir. Hangi tuşa basıldığını deşifre etmek için alıcı mikrodenetleyicinin uzaktan kumandadaki her bir tuşa hangi kodun karşılık geldiğini bilmesi gerekir.

Farklı uzaktan kumandalar, tuşa basma için farklı kodlar gönderir, bu nedenle kendi uzaktan kumandanızdaki her bir tuş için oluşturulan kodu belirlemeniz gerekir. Veri sayfasını bulabilirseniz, IR anahtar kodları listelenmelidir. Olmazsa, popüler uzaktan kumandaların çoğunu okuyacak ve bir tuşa bastığınızda onaltılık kodları seri monitöre yazdıracak basit bir Arduino çizimi var. Bunu nasıl kuracağınızı birazdan göstereceğim, ama önce alıcıyı Arduino'ya bağlamamız gerekiyor…

## ARDUİNO'YA IR ALICISI NASIL BAĞLANIR

Birkaç farklı IR alıcı türü vardır, bazıları bağımsızdır ve bazıları bir devre kartına monte edilmiştir. [Pinler, burada kullandığım HX1838 IR alıcısı ve uzaktan kumanda setinden](https://www.amazon.com/KOOBOOK-Infrared-Wireless-Control-Receiver/dp/B07S67SFSF/ref=as_li_ss_tl?dchild=1&keywords=arduino+ir+remote+control&qid=1590298098&sr=8-4&linkCode=ll1&tag=circbasi-20&linkId=52b8cbc41e25ce0cb2373cc678fd152f&language=en_US" \t "_blank) farklı düzenlenebileceğinden, özel IR alıcınızın veri sayfasını kontrol edin   . Ancak, tüm IR alıcılarının üç pimi olacaktır: sinyal, toprak ve Vcc.

Donanım bağlantılarıyla başlayalım. Çoğu koparma panosundaki pin düzeni şöyle görünür:

**[](file:///C:\Users\User\Desktop\)GND = “-“ akım**

**VCC = “+” akım**

**S = Gelen veriyi okuma**

Devre kartına monte edilmiş bir IR alıcısını bağlamak için Arduino'ya şu şekilde bağlayın:

## [Arduino IR Uzak Alıcı - Devre Kartı Bağlantı Şeması](file:///C:\Users\User\Desktop\)

### IRREMOTE KİTAPLIĞINI YÜKLEYİN

Aşağıdaki tüm kod örnekleri için IRremote kitaplığını kullanacağız. Kütüphanenin ZIP dosyasını buradan indirebilirsiniz .

<https://www.arduino.cc/reference/en/libraries/irremote/>

Kütüphaneyi ZIP dosyasından kurmak için Arduino IDE'yi açın, ardından Sketch > Include Library > Add .ZIP Library'ye gidin, ardından yukarıdaki bağlantıdan indirdiğiniz IRremote ZIP dosyasını seçin.

### KUMANDANIZIN KODLARINI BULUN

Uzaktan kumandanızın anahtar kodlarını bulmak için bu kodu Arduino'nuza yükleyin ve seri monitörü açın:

#include <IRremote.h>

const int RECV\_PIN = 7;

IRrecv irrecv(RECV\_PIN);

decode\_results results;

void setup(){

Serial.begin(9600);

irrecv.enableIRIn();

irrecv.blink13(true);

}

void loop(){

if (irrecv.decode(&results)){

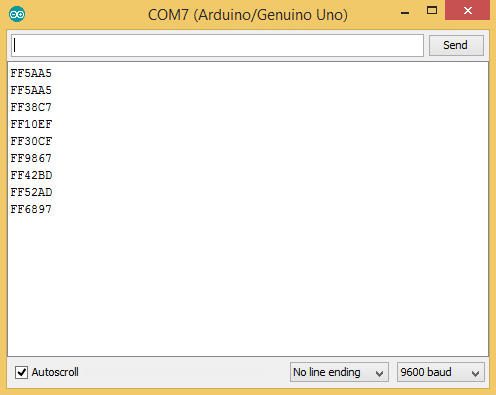
Serial.println(results.value, HEX);

irrecv.resume();

}

}

Şimdi uzaktan kumandanızdaki her tuşa basın ve her tuşa basıldığında yazdırılan onaltılık kodu kaydedin.

[](https://www.circuitbasics.com/wp-content/uploads/2017/05/Arduino-IR-Remote-Receiver-Finding-Remote-HEX-Codes.png)

Yukarıdaki programı kullanarak, HX1838 IR alıcım ve uzaktan kumanda setimle birlikte gelen uzaktan kumandadan bir anahtar tablosu ve bunlara karşılık gelen kodları çıkardım. Bir tuşa sürekli bastığınızda bir 0XFFFFFFFF kodu alacağınızı unutmayın.

## IR ALICISININ PROGRAMLANMASI

Alıcıyı bağladıktan sonra Arduino kütüphanesini kurabilir ve programlamaya başlayabiliriz. Aşağıdaki örneklerde size uzaktan kumandanız tarafından gönderilen kodları nasıl bulacağınızı, kumandanızın kullandığı IR protokolünü nasıl bulacağınızı, tuşa basışları seri monitöre veya LCD'ye nasıl yazdıracağınızı ve son olarak nasıl kontrol edeceğinizi göstereceğim. Arduino'nun çıkış pinleri bir uzaktan kumanda ile.

#include <IRremote.h>

const int RECV\_PIN = 7;

IRrecv irrecv(RECV\_PIN);

decode\_results results;

unsigned long key\_value = 0;

void setup(){

Serial.begin(9600);

irrecv.enableIRIn();

irrecv.blink13(true);

}

void loop(){

if (irrecv.decode(&results)){

if (results.value == 0XFFFFFFFF)

results.value = key\_value;

switch(results.value){

case 0xFFA25D:

Serial.println("1");

break;

case 0xFF629D:

Serial.println("2");

break;

case 0xFFE21D:

Serial.println("3");

break;

case 0xFF22DD:

Serial.println("4");

break;

case 0xFF02FD:

Serial.println("5");

break ;

case 0xFFC23D:

Serial.println("6");

break ;

case 0xFFE01F:

Serial.println("7");

break ;

case 0xFFA857:

Serial.println("8");

break ;

case 0xFF906F:

Serial.println("9");

break ;

case 0xFF9867:

Serial.println("0");

break ;

case 0xFF18E7:

Serial.println("up");

break ;

case 0xFF10EF:

Serial.println("left");

break ;

case 0xFF5AA5:

Serial.println("right");

break ;

case 0xFF42BD:

Serial.println("down");

break ;

}

key\_value = results.value;

irrecv.resume();

}

}

case 0xFFA25D:

Serial.println(“1“);

Bu satırlarda hex kodu 0xFFA25Dalındığında Arduino “1“ yazdırır.

### KOD NASIL ÇALIŞIR?

IRremote kitaplığını kullanan herhangi bir IR iletişimi için, öncelikle adlı bir nesne oluşturmamız irrecv ve IR alıcısının bağlı olduğu pin numarasını belirtmemiz gerekiyor (satır 3). Bu nesne, alıcıdan gelen bilgilerin protokolü ve işlenmesiyle ilgilenecektir.

Bir sonraki adım,  kodu çözülen bilgileri uygulamamızla paylaşmak için nesne tarafından kullanılacak sınıftan  resultsadlı bir nesne oluşturmaktır (satır 5).decode\_resultsirrecv

Blokta , void setup()önce seri monitör baud hızını yapılandırıyoruz. IRrecvArdından, üye işlevini çağırarak IR alıcısını başlatırız enableIRIn()(satır 10).

11.  satırdaki işlev, alıcı uzaktan kumandadan hata ayıklama için yararlı olan her sinyal aldığında Arduino'nun yerleşik LED'ini yanıp sönecektir.irrecv.blink13(true)

Blokta , bir kod alındığında void loop()işlev irrecv.decodetrue değerini döndürür ve program if ifadesindeki kodu yürütür. Alınan kod içinde saklanır  results.value. Sonra her bir IR kodunu işlemek ve karşılık gelen anahtar değerini yazdırmak için bir anahtar kullandım.

Anahtar bloğu başlamadan önce bir koşullu blok vardır:

if (results.value == 0XFFFFFFFF)

results.value = key\_value;

Kumandadan 0XFFFFFFFF alırsak bir önceki anahtarın tekrarı anlamına gelir. Bu nedenle, tekrarlama anahtarı modelini işlemek için, key\_value her kod alındığında onaltılı kodu global bir değişkende saklıyorum:

key\_value = results.value;

Bir tekrar deseni aldığınızda, daha önce kaydedilen değer geçerli tuşa basıldığında kullanılır.

Bölümün sonunda alıcıyı resetlemek ve bir sonraki kodu almaya hazırlamak için void loop()çağırıyoruz  .irrecv.resume();

## TUŞLARI LCD'YE YAZDIRIN

# Arduino 16×2 LCD Ekran Kullanımı

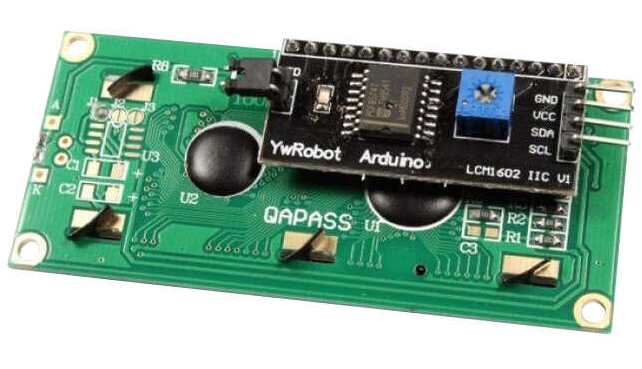
Arduino 16×2 LCD Ekran kullanımı, birçok projede kullanıcı arayüzü bakımında oldukça zengin bir kullanım alanına sahiptir. LCD ekran kullanarak, veri takibi, basit oyun senaryoları, güvenlik sistemleri, sensör okumaları ya da otomasyon işlemlerine entegre edebilirsiniz.

16x2 Karakter LCD ekranların bazı modelleri üzerinde I2C devresi ile beraber gelmektedir bu sayede kalabalık kablo görüntüsü yerine 2 adet kablo ile veri aktarımı sağlanmaktadır I2C entegresi üzerinde gelen bir LCD ekranla hem de I2C devresi olmayan bir LCD ekran ile işlem yapacağız.

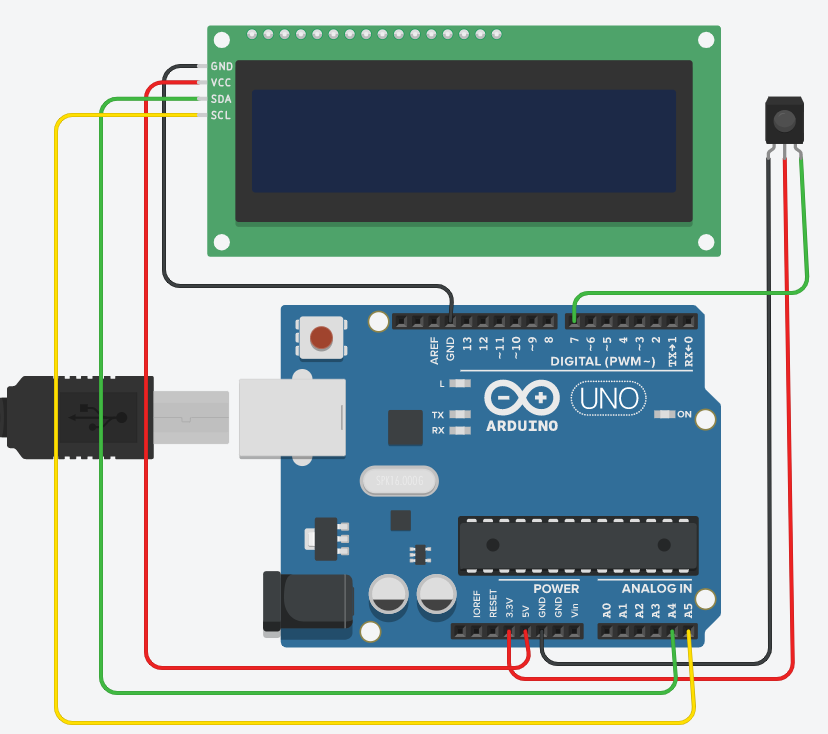
## I2C Devresi Olan LCD Ekran

I2C, Inter-IC’nin kısaltmasıdır. Ve bir BUS türüdür. I2C, senkron, çoklu bağımlı, çoklu ana paket anahtarlamalı, tek uçlu bir seri veriyoludur. Aynı veriyoluna birden fazla çip bağlanabilir, Seri Veri Hattı (SDA) ve Seri Saat Hattı (SCL) kullanır. Diğer voltajlara sahip sistemlere izin verilmesine rağmen, kullanılan tipik voltajlar +5 V veya +3,3 V’dir.

## I2C Devresi Olan LCD Ekran



Temel bağlantı şeması şöyle görünür:



Ekran Çıktısı Almak için:

Arduino IDE ile önceden yüklenmiş olarak gelen **<LiquidCrystal\_I2C.h>** kitaplığını kullanır. Kitaplık, bir programa kısaltılmış biçimde kolayca eklenebilen bir dizi işlevdir.

Aşağıdaki koddaki 1. satır bunu komutla yapar  #include <LiquidCrystal.h>. Bir programa bir kütüphane eklediğinizde, kütüphanedeki tüm kodlar programınızın koduyla birlikte Arduino'ya yüklenir.

Şimdi programlamaya girmeye hazırız! Birazdan yapabileceğiniz daha ilginç şeylerin üzerinden geçeceğim, ama şimdilik sadece basit bir test programı çalıştıralım. Bu program “merhaba, dünya!” yazdıracak. ekrana. Bu kodu Arduino IDE'ye girin ve panoya yükleyin:

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {

lcd.begin(16, 2);

lcd.print("hello, world!");

}

void loop() {

}

LCD ekranınız şöyle görünmelidir:

**Resim Tarafımızca çekilmiştir.**



## LCD EKRAN SEÇENEKLERİ

LiquidCrystal kitaplığında kullanabileceğimiz 19 farklı fonksiyon bulunmaktadır. Bu işlevler, metnin konumunu değiştirmek, metni ekran boyunca taşımak veya ekranı açıp kapatmak gibi şeyler yapar. Aşağıda, her bir işlevin kısa bir açıklaması ve bir programda nasıl kullanılacağı yer almaktadır.

**lcd.begin()**

Bu fonksiyon LCD'nin boyutlarını ayarlar. void setup()Program bölümündeki herhangi bir LiquidCrystal işlevinden önce yerleştirilmesi gerekir. Satır ve sütun sayısı olarak belirtilir lcd.begin(columns, rows). 16×2 LCD lcd.begin(16, 2)için kullanırsınız.

**lcd.clear()**

Bu işlev, LCD'de halihazırda görüntülenen herhangi bir metni veya veriyi siler. lcd.clear()

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(12, 11, 5, 4, 3, 2);

void setup() {

lcd.begin(16, 2);

}

void loop() {

lcd.print("hello, world!");

lcd.clear();

}

**lcd.print()**

Bu işlev, metni LCD'ye yazdırmak için kullanılır. void setup()Programın bölümünde veya bölümünde kullanılabilir void loop().

Harfleri ve sözcükleri yazdırmak için metnin etrafına tırnak işaretleri (” “) koyun. Örneğin, *merhaba, dünya!*, kullanın lcd.print("hello, world!").

**lcd.setCursor()**

Bu işlev, imleci (ve herhangi bir basılı metni) ekrandaki herhangi bir konuma yerleştirir. void setup()Programınızın bölümünde veya void loop()bölümünde kullanılabilir .

İmleç konumu ile tanımlanır  lcd.setCursor(column, row). Sütun ve satır koordinatları sıfırdan başlar (sırasıyla 0-15 ve 0-1). Örneğin,  “merhaba, dünya!” bölümünde kullanmak lcd.setCursor(2, 1). void setup()yukarıdaki program “merhaba dünya!” yazdırır. alt çizgiye kaydırır ve onu iki boşluk sağa kaydırır:

Resim Tarafımızca çekilmiştir.



Kodları:

#include <LiquidCrystal.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {

lcd.begin(16, 2);

lcd.setCursor(2, 1);

lcd.print("hello, world!");

}

void loop() {

}

 Gelelim kumandamızdan basılan tuşları LCD’de yazdırmamıza:

Alıcımız için oluşturduğumuz switch-case yapısının içersinine yukarıda anlattığım içerikleri ekliyoruz:

#include <IRremote.h>

#include <LiquidCrystal\_I2C.h>

const int RECV\_PIN = 7;

IRrecv irrecv(RECV\_PIN);

decode\_results results;

unsigned long key\_value = 0;

void setup(){

Serial.begin(9600);

irrecv.enableIRIn();

irrecv.blink13(true);

}

void loop(){

if (irrecv.decode(&results)){

if (results.value == 0XFFFFFFFF)

results.value = key\_value;

switch(results.value)

{

case 0xFFA25D:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : 1");

Serial.print("1");

break;

case 0xFF629D:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : 2");

Serial.print("2");

break;

case 0xFFE21D:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : 3");

Serial.print("3");

break;

case 0xFF22DD:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : 4");

Serial.print("4");

break;

case 0xFF02FD:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : 5");

Serial.print("5");

break ;

case 0xFFC23D:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : 6");

Serial.print("6");

break ;

case 0xFFE01F:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : 7");

Serial.print("7");

break ;

case 0xFFA857:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : 8");

Serial.print("8");

break ;

case 0xFF906F:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : 9");

Serial.print("9");

break ;

case 0xFF9867:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : 0");

Serial.print("0");

break ;

case 0xFF10EF:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : left");

Serial.print("left");

break;

case 0xFF5AA5:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : right");

Serial.print("right");

break;

case 0xFF4AB5:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : down");

Serial.print("down");

break ;

case 0xFF18E7:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : up");

Serial.print("up");

break;

case 0xFF6897:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : #");

Serial.print("#");

break ;

case 0xFF38C7:

lcd.clear();

lcd.setCursor(0,0); // İlk satırın başlangıç noktası

lcd.print("Basilan");

lcd.setCursor(0,1); // İkinci satırın başlangıç noktası

lcd.print("Tus : OK");

Serial.print("OK");

break ;

case 0xFFB04F:

lcd.clear();

Serial.print("\*");

break;

}

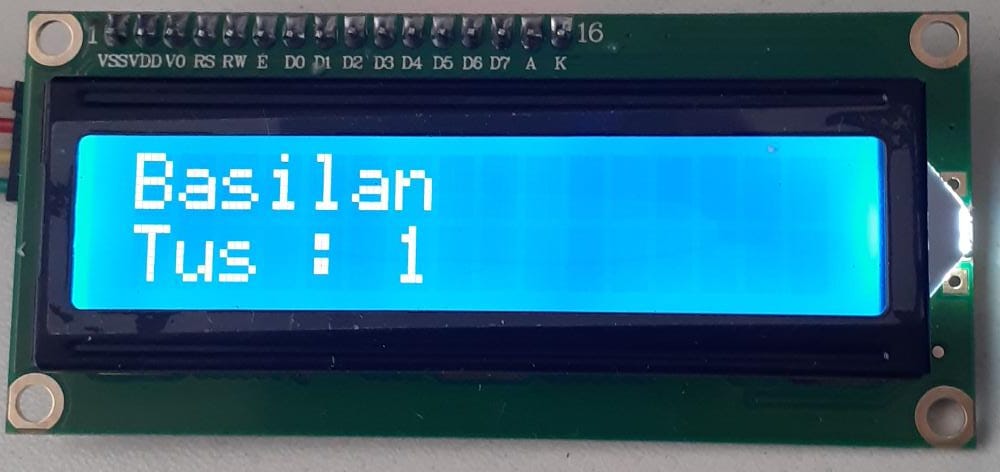
key\_value = results.value;

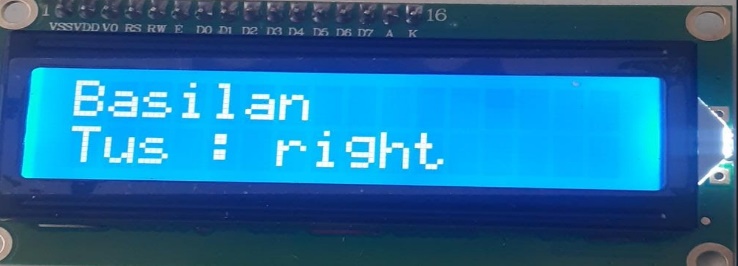
irrecv.resume();

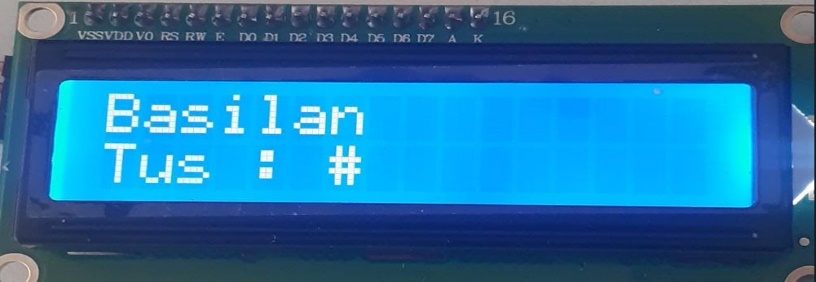
}

}

Ekranımıza gelen sonuç böyledir:

****





**System.IO.Ports**

**Seri bağlantı noktalarını denetlemek için sınıflar içerir. En önemli sınıf, SerialPort zaman uyumlu ve olay odaklı g/ç için bir çerçeve, PIN ve kesme durumlarına erişim ve seri sürücü özelliklerine erişim sağlar. Nesneleri kaydırmak için kullanılabilir ve bu Stream , akış kullanan sınıfların seri bağlantı noktasına erişimine izin verir.Bu kütüphane sayesinde “COM” portuna USB kablo ile bağlı olan arduniomuzdan gelen verileri işleyebileceğiz.**

**System.Threading**

**Bir işlem başlatıldığında, ortak dil çalışma zamanı otomatik olarak uygulama kodunu yürütmek için tek bir ön plan iş parçacığı oluşturur. Bu ana ön plan iş parçacığıyla birlikte bir işlem, işlemle ilişkili program kodunun bir bölümünü yürütmek için bir veya daha fazla iş parçacığı oluşturabilir. Bu iş parçacıkları ön planda veya arka planda yürütülebilir. Ayrıca, ortak dil çalışma zamanı tarafından yönetilen çalışan iş parçacıklarında kod yürütmek için sınıfını kullanabilirsiniz.Bu kütüphane sayesinde fonksiyonlarımıza ms cinsinden gecikme yada bekleme süresi verebileceğiz.**

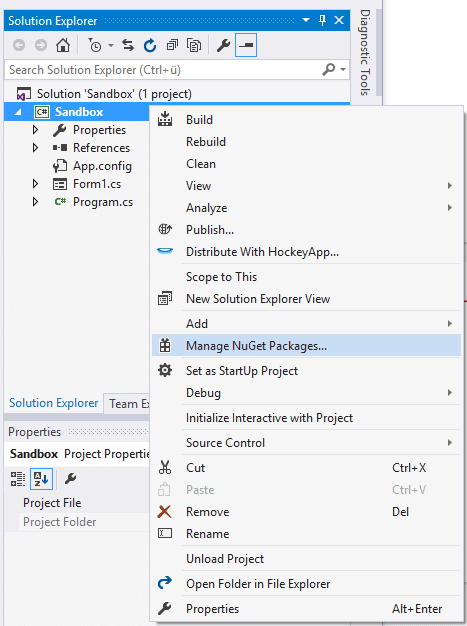
**Winforms'da C# ile InputSimulator kullanarak tuşa basmayı doğru şekilde**

**Windows Forms'daki harika InputSimulator kitaplığını kullanarak klavyedeki herhangi bir tuştan tuşa basma olayını nasıl simüle edeceğinizi öğrenin.**

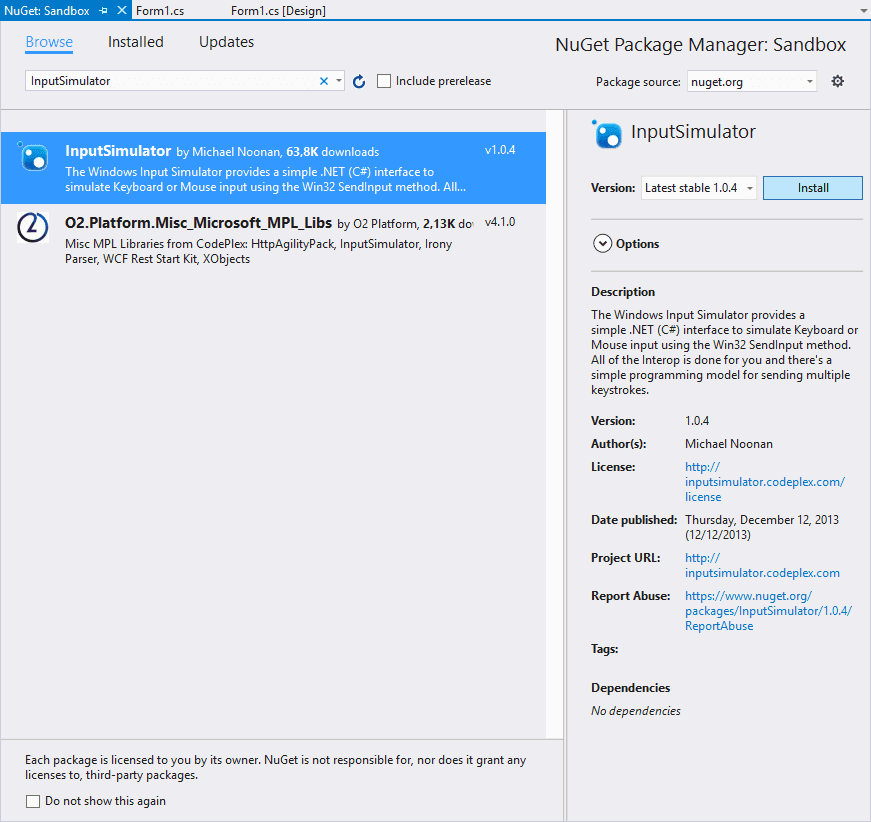
**Elbette Winforms ile çalışan tüm C# geliştiricileri, kodu bildiğiniz sürece herhangi bir tuşa basılmasını simüle etmeye izin veren parçacıkları bilir:**

**Ancak bu korumak için bir baş ağrısı haline gelebilir. Bu nedenle bu makalede, InputSimulator kullanarak tuşa basma olaylarını simüle etmenin doğru yolunu göstereceğiz. Windows Giriş Simülatörü, Win32 SendInput yöntemini kullanarak Klavye veya Fare girişini simüle etmek için basit bir .NET (C#) arabirimi sağlar. Birlikte Çalışmanın tamamı sizin için yapılır ve birden çok tuş vuruşu göndermek için basit bir programlama modeli vardır.**

**InputSimulator ile çalışmanın avantajları Windows Forms, metin girişini simüle edebilen ancak gerçek tuş vuruşlarını olmayan SendKeys yöntemini sağlar. Windows Giriş Simülatörü, Control, Alt, Shift, Tab, Enter, Space, Backspace, Windows Tuşu, Caps Lock, Num Lock, Scroll Lock, Volume dahil olmak üzere herhangi bir Klavye girişini sentezlemek veya simüle etmek için WPF, Windows Forms ve Konsol Uygulamalarında kullanılabilir. Yukarı/Aşağı ve Sessiz, Web, Posta, Arama, Sık Kullanılanlar, İşlev Tuşları, Geri ve İleri gezinme tuşları, Programlanabilir tuşlar ve Sanal Anahtar tablosunda tanımlanan diğer tuşlar. Metin girişi, aşağı tuş, yukarı tuş, tuşa basma ve karmaşık değiştirilmiş tuş vuruşlarını ve akorları simüle etmek için basit bir API sağlar. Bu kitaplığı kullanmaya başlamak ister misiniz? O halde okumaya devam edin! 1. InputSimulator'ı kurun Tuşa basma ve tuş vuruşlarını kolayca simüle etmek için ihtiyacınız olan ilk şey, kitaplığı projenize nuGet aracılığıyla kurmaktır. Winforms C# projenizi açın ve çözüm gezgininde NuGet paket yöneticisini açın:**



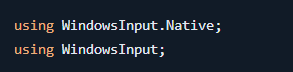
**Gözat sekmesine gidin ve şunu arayın** InputSimulator:



**Listeden Michael Noonan'ın paketini seçin ve kurun. Paketin kurulumu tamamlandıktan sonra sınıflarınızda kullanabileceksiniz. Bu kütüphane hakkında daha fazla bilgi ve örnekler için lütfen buradan Github'daki resmi depoyu ziyaret edin .**

[**https://github.com/michaelnoonan/inputsimulator**](https://github.com/michaelnoonan/inputsimulator)

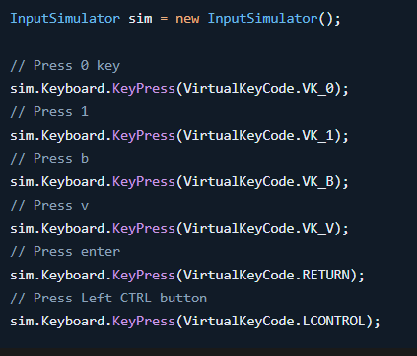
**2. InputSimulator'ı Kullanma**

****

**Sanal kod listesi Herhangi bir tuşun keypress olayını simüle etmek istiyorsanız , bu listeden alabileceğiniz keycode'a ihtiyacınız olacak . Ancak, WindowsInput.Native.VirtualKeyCodeher anahtar için bir özelliği olan enum özelliğini kullanabilirsiniz.**

**Tuşa Basmayı Simüle Et Tek bir tuşa basma olayını** Keyboard.KeyPress **simüle etmek için simüle etmek istediğiniz anahtarın sanal anahtar kodunu bekleyen yöntemi kullanın:**

**Kütüphanemizin örnek kodları:**

****

**Bizim kodlarımız:**

using System;

using System.IO.Ports;

using System.Threading;

using WindowsInput.Native;

using WindowsInput;

public static class Program

{

public static SerialPort \_serialPort;

public static void Main(string[] args)

{

InputSimulator sim = new InputSimulator();

Console.ForegroundColor = ConsoleColor.Green;

\_serialPort = new SerialPort();

\_serialPort.PortName = "COM9 //COM9 PORTUNU KULLANARAK BAĞLANIYORUZ.

\_serialPort.BaudRate = 9600;//9600 BAUD HIZINDA BAĞLANIYORUZ

\_serialPort.DataBits = 8; //Bayt başına standart veri biti uzunluğunu alır veya ayarlar.

\_serialPort.StopBits = StopBits.One; //Bayt başına standart stopbit sayısını alır veya ayarlar.

\_serialPort.DtrEnable = true; // Seri iletişim sırasında Veri Terminali Hazır (DTR) sinyalini etkinleştiren bir değer alır veya ayarlar.

\_serialPort.RtsEnable = true; // Gönderme İsteği (RTS) sinyalinin seri iletişim sırasında etkinleştirilip etkinleştirilmediğini belirten bir değer alır veya ayarlar.

\_serialPort.Open();

while (\_serialPort.IsOpen) ///Sürekli Nesnenin açık durumunu SerialPort gösteren bir değer alır.

{

string a = \_serialPort.ReadExisting();// Hem akışta hem de nesnenin giriş arabelleğinde kodlamaya göre hemen kullanılabilir olan tüm baytları SerialPort okur.

if (a == "1") // arduniodaki HEX değerin karşılığı "1" ise

{

sim.Mouse.LeftButtonClick();//Mouse'de sol tıklama yap

sim.Mouse.MoveMouseTo(11, 22); // "X" ve "Y" eksenindeki belirlenmiş koordinatlara mouse'yi taşır.

//Mouse sol tıklama ve belirlediğimiz koordinata sürükleme işlemini aynı anda yaptırıyoruz.

}

else if (a == "2") // arduniodaki HEX değerin karşılığı "2" ise

{

int i = 0; //indexsimizi 0 yaptık

while (i < 10) // 10 dan küçük oluncaya kadar devam etmesini sağladık

{

sim.Keyboard.KeyPress(VirtualKeyCode.VK\_W); / index 10'dan küçük oluncaya kadar W tuşuna basmasını sağlıyoruz

Thread.Sleep(1); // 1ms gecikmeli olarak yapmasını sağlıyoruz

i++; // i yi 1 kez arttırıyor.

}

}

else if (a == "3") // arduniodaki HEX değerin karşılığı "3" ise

{

sim.Keyboard.KeyPress(VirtualKeyCode.SPACE);

//"SPACE" tuşuna basmasını sağlıyoruz

}

}

else if (a == "4") // arduniodaki HEX değerin karşılığı "4" ise

{

int i = 0;

while (i <= 10)

{

sim.Keyboard.KeyPress(VirtualKeyCode.VK\_A); // index 10'dan küçük oluncaya kadar "A" tuşuna basmasını sağlıyoruz

Thread.Sleep(1); // 1ms gecikmeli olarak yapmasını sağlıyoruz

i++; // i yi 1 kez arttırıyor.

}

}

else if (a == "5") // arduniodaki HEX değerin karşılığı "5" ise

{

sim.Keyboard.KeyPress(VirtualKeyCode.VK\_G); //"SPACE" tuşuna basmasını sağlıyoruz

}

else if (a == "6")// arduniodaki HEX değerin karşılığı "6" ise

{

int i = 0;

while (i < 10)

{

sim.Keyboard.KeyPress(VirtualKeyCode.VK\_D); // index 10'dan küçük oluncaya kadar "D" tuşuna basmasını sağlıyoruz

Thread.Sleep(1); // 1ms gecikmeli olarak yapmasını sağlıyoruz

i++;i yi 1 kez arttırıyor.

}

}

else if (a == "7")

{

sim.Keyboard.KeyPress(VirtualKeyCode.VK\_1); // Klavyeden "1" tuşuna basıyor.(NUMPAD 1 tuşu değil).

sim.Mouse.LeftButtonClick();//Mouse'de sol tıklama yapıyor.

//Mouse sol tıklama ve "1" tuşuna basma işlemini aynı anda yaptırıyoruz. }

else if (a == "8")// arduniodaki HEX değerin karşılığı "8" ise

{

int i = 0;

while (i < 10)

{

sim.Keyboard.KeyPress(VirtualKeyCode.VK\_S); index 10'dan küçük oluncaya kadar "S" tuşuna basmasını sağlıyoruz

Thread.Sleep(1); // 1ms gecikmeli olarak yapmasını sağlıyoruz

i++;i yi 1 kez arttırıyor.

}

}

else if (a == "9")

{

sim.Keyboard.KeyPress(VirtualKeyCode.VK\_2); ;// Klavyeden "2" tuşuna basıyor.(NUMPAD 2 tuşu değil).

sim.Mouse.LeftButtonClick();//Mouse'de sol tıklama yapıyor.

//Mouse sol tıklama ve "2" tuşuna basma işlemini aynı anda yaptırıyoruz.

}

else if (a == "0") // arduniodaki HEX değerin karşılığı "0" ise

{

sim.Keyboard.KeyPress(VirtualKeyCode.VK\_3); // Klavyeden "3" tuşuna basıyor.(NUMPAD 3 tuşu değil).

sim.Mouse.LeftButtonClick();//Mouse'de sol tıklama yapıyor.

//Mouse sol tıklama ve "3" tuşuna basma işlemini aynı anda yaptırıyoruz.

}

else if (a == "OK")// arduniodaki HEX değerin karşılığı "OK" ise

{

// şuanlık boş bırakıldı

}

else if (a == "right")// arduniodaki HEX değerin karşılığı "right" ise

{

// x eksenindeki 1920 pixelde 65535.0f hücre vardır.(65535.0f \* (1750 / (float)1920)) + 0.5f) işlemini yaptırdığımızda x ekseninin ortasına yakın yeri buluyoruz

int x = (int)((65535.0f \* (1750 / (float)1920)) + 0.5f);

// y eksenindeki 1080 pixelde 65535.0f hücre vardır.(65535.0f \* (1750 / (float)1080)) + 0.5f) işlemini yaptırdığımızda y ekseninin ortasına yakın yeri buluyoruz

int y = (int)((65535.0f \* (850 / (float)1080)) + 0.5f);

sim.Mouse.MoveMouseTo(x, y); // ve bulduğumuz "X" , "Y" eksenlerine mousemizi taşıyoruz.

}

else if (a == "left")// arduniodaki HEX değerin karşılığı "left" ise

{

// x eksenindeki 1920 pixelde 65535.0f hücre vardır.(65535.0f \* (1750 / (float)1920)) + 0.5f) işlemini yaptırdığımızda x ekseninin ortasına yakın yeri buluyoruz

int x = (int)((65535.0f \* (1500 / (float)1920)) + 0.5f);

// y eksenindeki 1920 pixelde 65535.0f hücre vardır.(65535.0f \* (356 / (float)1920)) + 0.5f) işlemini yaptırdığımızda y ekseninin ortasına yakın yeri buluyoruz

int y = (int)((65535.0f \* (356 / (float)1080)) + 0.5f);

sim.Mouse.MoveMouseTo(x, y); ve bulduğumuz "X" , "Y" eksenlerine mousemizi taşıyoruz.

}

else if (a == "up")

{

// x eksenindeki 1920 pixelde 65535.0f hücre vardır.(65535.0f \* (236 / (float)1920)) + 0.5f) işlemini yaptırdığımızda x ekseninin ortasına yakın yeri buluyoruz

int x = (int)((65535.0f \* (236 / (float)1920)) + 0.5f);

// y eksenindeki 1920 pixelde 65535.0f hücre vardır.(65535.0f \* (865 / (float)1920)) + 0.5f) işlemini yaptırdığımızda y ekseninin ortasına yakın yeri buluyoruz

int y = (int)((65535.0f \* (865 / (float)1080)) + 0.5f);

sim.Mouse.MoveMouseTo(x, y); // ve bulduğumuz "X" , "Y" eksenlerine mousemizi taşıyoruz.

}

else if (a == "down")

{

// x eksenindeki 1920 pixelde 65535.0f hücre vardır.(65535.0f \* (250 / (float)1920)) + 0.5f) işlemini yaptırdığımızda x ekseninin ortasına yakın yeri buluyoruz

int x = (int)((65535.0f \* (250/(float)1920)) + 0.5f);

// y eksenindeki 1920 pixelde 65535.0f hücre vardır.(65535.0f \* (750 / (float)1920)) + 0.5f) işlemini yaptırdığımızda y ekseninin ortasına yakın yeri buluyoruz

int y = (int)((65535.0f \* (750/(float)1080)) + 0.5f);

sim.Mouse.MoveMouseTo(x, y); // ve bulduğumuz "X" , "Y" eksenlerine mousemizi taşıyoruz.

}

}

}

**Oyun içi ekran görüntüleri**

**Counter Strike 1.6 De\_dust2 haritasında test edilmiştir.**

**Bazı oyunların içerisinde “Anti Cheat” yazılımları bulunduğundan dolayı projemizi zararlı yazılım olarak algılayabilir ve çalışmayabilir yada Hesabınız engellebilir!**

**SORUMLULUK TAMAMEN KULLANICIYA AİTTİR!!!**

**Standart oyun içi açısı:**

****

**“1” numaralı tuşa basıldığında (sim.Mouse.LeftButtonClick()) mousemizin sol tuşuna tıklar.Bu oyun içi ateş etmemizi sağlar.**

**“x” ve “y” ekseninde**  (**sim.Mouse.MoveMouseTo(11, 22))**

**x=11,y=22; eksenine mousemizi taşır.**

**Bu fonksiyonların ikisinide aynı anda yapar.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

**“2” numaralı tuşa basıldığında 10 defa “W” tuşuna basar ve oyun içi ileri yönde yürümemizi sağlar**

**sim.Keyboard.KeyPress(VirtualKeyCode.VK\_W); fonksiyonu ile sağlanır.**

**Thread.Sleep(1); fonksiyonu ile “1 ms” aralıkla tek tek “W” tuşuna basmayı sağlar.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

**Standart oyun içi açısı resim ile karşılaştırabilirsiniz.**

****

**“3” numaralı tuşa basıldığında “SPACE” tuşuna basar ve oyun içindeki karakterimizin zıplamasını sağlar.**

**sim.Keyboard.KeyPress(VirtualKeyCode.SPACE); fonksiyonu ile çalışır.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

**Resimde görüldüğü gibi karakterimizin ayakları yere temas etmiyor ve gölgesi zeminde görünüyor.**

**“4” numaralı tuşa basıldığında 10 defa “A” tuşuna basar ve oyun içi sol yönde yürümemizi sağlar.**

**sim.Keyboard.KeyPress(VirtualKeyCode.VK\_A); fonksiyonu ile sağlanır.**

**Thread.Sleep(1); fonksiyonu ile “1 ms” aralıkla tek tek “A” tuşuna basmayı sağlar.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

**Standart oyun içi açısı resim ile karşılaştırabilirsiniz.**

****

**“5” numaralı tuşa basıldığında “G” tuşuna basar.Oyun içi karakterimizin silahını yere atmasını sağlar.**

**sim.Keyboard.KeyPress(VirtualKeyCode.VK\_G); fonksiyonu ile sağlar.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

**Aşağıdaki resimde kırmızı ile işaretlemiş alan içerisinde karakterimizin yere attığı silah mevcuttur.**

**“6” numaralı tuşa basıldığında 10 defa “D” tuşuna basar ve oyun içi sol yönde yürümemizi sağlar.**

**sim.Keyboard.KeyPress(VirtualKeyCode.VK\_D); fonksiyonu ile sağlanır.**

**Thread.Sleep(1); fonksiyonu ile “1 ms” aralıkla tek tek “A” tuşuna basmayı sağlar.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

**Standart oyun içi açısı resim ile karşılaştırabilirsiniz.**

**“7” numaralı tuşa basıldığında karakterimizin 1.envanterindeki silahına(eğer mevcutsa) geçiş yapar.**

**sim.Keyboard.KeyPress(VirtualKeyCode.VK\_1); fonksiyonu ile “1” tuşuna basar(NUMPAD 1 tuşu değil!).**

**sim.Mouse.LeftButtonClick();fonksiyonu ile Mouse sol tuşuna basar ve silahını seçer.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

****

**“8” numaralı tuşa basıldığında 10 defa “S” tuşuna basar ve oyun içi ileri yönde yürümemizi sağlar**

**sim.Keyboard.KeyPress(VirtualKeyCode.VK\_S); fonksiyonu ile sağlanır.**

**Thread.Sleep(1); fonksiyonu ile “1 ms” aralıkla tek tek “S” tuşuna basmayı sağlar.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

**Standart oyun içi açısı resim ile karşılaştırabilirsiniz.**

****

**“9” numaralı tuşa basıldığında karakterimizin 2.envanterindeki silahına(eğer mevcutsa) geçiş yapar.**

**sim.Keyboard.KeyPress(VirtualKeyCode.VK\_2); fonksiyonu ile “2” tuşuna basar(NUMPAD 2 tuşu değil!).**

**sim.Mouse.LeftButtonClick();fonksiyonu ile Mouse sol tuşuna basar ve silahını seçer.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

****

**“0” numaralı tuşa basıldığında karakterimizin 3.envanterindeki silahına(eğer mevcutsa) geçiş yapar.**

**sim.Keyboard.KeyPress(VirtualKeyCode.VK\_3); fonksiyonu ile “3” tuşuna basar(NUMPAD 3 tuşu değil!).**

**sim.Mouse.LeftButtonClick();fonksiyonu ile Mouse sol tuşuna basar ve silahını seçer.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

****

**“UP” tuşuna (yukarı ok tuşu) basıldığında**  **x eksenindeki 1920 pixelde 65535.0f hücre vardır** int x = (int)((65535.0f \* (1750 / (float)1920)) + 0.5f); **işlemini yaptırdığımızda x ekseninin ortasına yakın yeri buluyoruz .Y eksenindede aynı şekilde 1080 pixelde 65535.0f hücre vardır** int y = (int)((65535.0f \* (850 / (float)1080)) + 0.5f); **işlemini yaptırdığımızda y ekseninin ortasına yakın yeri buluyoruz.**

**sim.Mouse.MoveMouseTo(x, y); fonksiyonu ile bulduğumuz koordinatlara mousemizi sürüklüyoruz.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

**Standart oyun içi açısı resime göre mousemizi biraz sol,capraz aşağıya sürüklüyor.**

**İsteğe göre bu koordinatlar değiştirelebilir.**

**“LEFT” tuşuna (Sol ok tuşu) basıldığında**  **x eksenindeki 1920 pixelde 65535.0f hücre vardır** int x = (int)((65535.0f \* (1500/ (float)1920)) + 0.5f); **işlemini yaptırdığımızda x ekseninin ortasına yakın yeri buluyoruz .Y eksenindede aynı şekilde 1080 pixelde 65535.0f hücre vardır** int y = (int)((65535.0f \* (356/ (float)1080)) + 0.5f); **işlemini yaptırdığımızda y ekseninin ortasına yakın yeri buluyoruz.**

**sim.Mouse.MoveMouseTo(x, y); fonksiyonu ile bulduğumuz koordinatlara mousemizi sürüklüyoruz.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

**Standart oyun içi açısı resime göre mousemizi biraz sağ,capraz yukarıya sürüklüyor.**

**İsteğe göre bu koordinatlar değiştirelebilir.**

**“DOWN” tuşuna (aşağı ok tuşu) basıldığında**  **x eksenindeki 1920 pixelde 65535.0f hücre vardır** int x = (int)((65535.0f \* (250/ (float)1920)) + 0.5f); **işlemini yaptırdığımızda x ekseninin ortasına yakın yeri buluyoruz .Y eksenindede aynı şekilde 1080 pixelde 65535.0f hücre vardır** int y = (int)((65535.0f \* (750/ (float)1080)) + 0.5f); **işlemini yaptırdığımızda y ekseninin ortasına yakın yeri buluyoruz.**

**sim.Mouse.MoveMouseTo(x, y); fonksiyonu ile bulduğumuz koordinatlara mousemizi sürüklüyoruz.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

**Standart oyun içi açısı resime göre mousemizi biraz sol,capraz aşağıya sürüklüyor.**

**İsteğe göre bu koordinatlar değiştirelebilir.**

****

**“RİGHT” tuşuna (sağ ok tuşu) basıldığında**  **x eksenindeki 1920 pixelde 65535.0f hücre vardır** int x = (int)((65535.0f \* (1750/ (float)1920)) + 0.5f); **işlemini yaptırdığımızda x ekseninin ortasına yakın yeri buluyoruz .Y eksenindede aynı şekilde 1080 pixelde 65535.0f hücre vardır** int y = (int)((65535.0f \* (850/ (float)1080)) + 0.5f); **işlemini yaptırdığımızda y ekseninin ortasına yakın yeri buluyoruz.**

**sim.Mouse.MoveMouseTo(x, y); fonksiyonu ile bulduğumuz koordinatlara mousemizi sürüklüyoruz.**

**Tuşa basılı tutulduğunda sürekli devam eder.**

**Standart oyun içi açısı resime göre mousemizi biraz sağ,capraz yukarı sürüklüyor.**

**İsteğe göre bu koordinatlar değiştirelebilir.**



**Kaynakça**

[**https://www.makerguides.com/ir-receiver-remote-arduino-tutorial/**](https://www.makerguides.com/ir-receiver-remote-arduino-tutorial/)

[**https://create.arduino.cc/projecthub/Nicholas\_N/control-an-led-with-the-remote-control-79b6fd**](https://create.arduino.cc/projecthub/Nicholas_N/control-an-led-with-the-remote-control-79b6fd)

[**https://create.arduino.cc/projecthub/electropeak/use-an-ir-remote-transmitter-and-receiver-with-arduino-1e6bc8**](https://create.arduino.cc/projecthub/electropeak/use-an-ir-remote-transmitter-and-receiver-with-arduino-1e6bc8)

[**https://randomnerdtutorials.com/arduino-ir-remote-control/**](https://randomnerdtutorials.com/arduino-ir-remote-control/)

[**https://steemit.com/utopian-io/@drencolha/arduino-irremote-library-kurulumu-ve-kullanimi-turkish-tutorial**](https://steemit.com/utopian-io/@drencolha/arduino-irremote-library-kurulumu-ve-kullanimi-turkish-tutorial)

[**https://akademi.robolinkmarket.com/16x2-lcd-ekran-ve-i2c-modulu-kullanimi/**](https://akademi.robolinkmarket.com/16x2-lcd-ekran-ve-i2c-modulu-kullanimi/)

[**https://www.projehocam.com/arduino-lcd-i2c-protokolu-kullanimi/**](https://www.projehocam.com/arduino-lcd-i2c-protokolu-kullanimi/)

[**https://ourcodeworld.com/articles/read/520/simulating-keypress-in-the-right-way-using-inputsimulator-with-csharp-in-winforms**](https://ourcodeworld.com/articles/read/520/simulating-keypress-in-the-right-way-using-inputsimulator-with-csharp-in-winforms)

[**https://www.geeksforgeeks.org/how-to-create-threads-in-c-sharp/#:~:text=In%20C%23%2C%20a%20multi%2Dthreading,Threading%20namespace**](https://www.geeksforgeeks.org/how-to-create-threads-in-c-sharp/#:~:text=In%20C%23%2C%20a%20multi%2Dthreading,Threading%20namespace)**.**

[**https://localcoder.org/using-system-io-ports-serialport-in-net-core-1-1**](https://localcoder.org/using-system-io-ports-serialport-in-net-core-1-1)