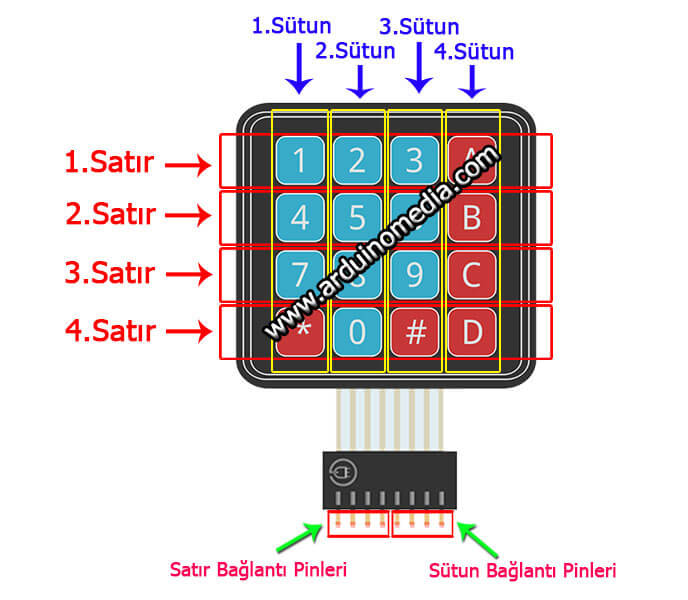
**Arduino ile 4×4 Membran Tuş Takımı Keypad Kullanımı – Uygulaması**

## Keypad Nedir?

Keypad birden fazla tuşun tek modül üzerinde birleştirilmiş halidir. [Arduino](https://www.arduinomedia.com/category/arduino/" \t "_blank) ile şifre girişi gerektiren ya da dışarıdan sayı girişi gerektiren arduino projelerinde kullanılmaktadır.

Keypadler satır ve sütun sayısına göre 4×3 veya 4×4 keypad olarak çeşitleri vardır. Bu derste **arduino** **4×4 keypad** kullanımını inceleyeceğiz. Fakat elinizde 4×3 keypad kullanımı da hemen hemen benzer yapıdadır.

4×4 keypad isminden de anlaşılacağı üzere 4 satır ve 4 sütundan oluşan tuş takımlarıdır.



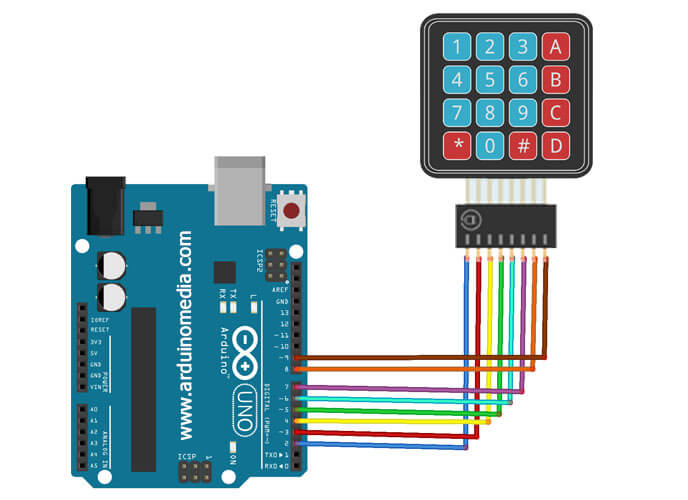
arduino-keypad-kullanımı

Arduino ile keypad kodları yazılırken bu satır ve sütunlarda bulunan karakterler yukarıdaki gibi tanımlanmalıdır.

**Keypad arduino bağlantısı** yapılırken soldaki 4 pin satır pinleri, sağdaki 4 pin ise sütun pinleri olarak belirtilmelidir.

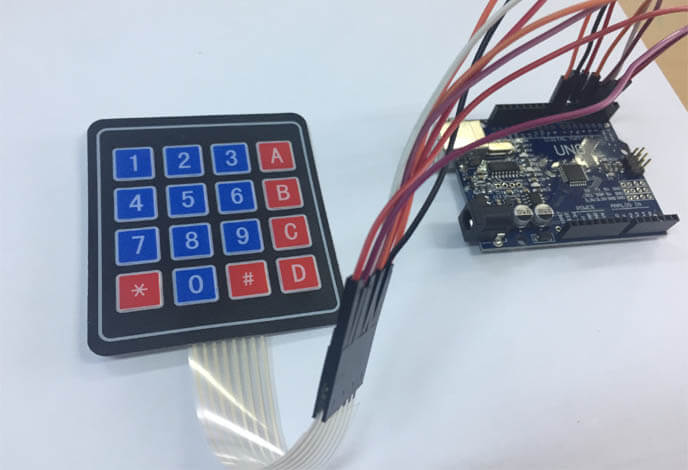
## ****Keypad Arduino Bağlantısı Nasıl Yapılır?****

Keypad arduino bağlantısı yapılırken keypad bağlantı pinleri arduino dijital pinlerine bağlanmalıdır. Keypad tuşları size doğru dönük iken soldaki 4 pin satır pinleri, sağdaki 4 pin ise sütun pinleridir. Yapacağımız örnek **arduino  keypad uygulaması** için satır pinlerini sırasıyla 2,3,4 ve 5 numaralı arduino pinlerine, sütun pinlerini ise 6,7,8 ve 9 numaralı arduino dijital pinlerine bağlayacağız.



arduino-ile-4×4-keypad-bağlantı-şeması

**Keypad arduino devresi**ni görseldeki şemayı inceleyerek hazırlayınız.



keypad-arduino-devresi

Şimdi **arduino ile keypad uygulaması** için gerekli **arduino keypad kodları**nı hazırlayalım.

## Keypad Tuş Takımı Arduino Kodları

Arduino ile keypad kullanabilmek için ayrıca bir kütüphane dosyasına ihtiyacımız vardır. Keypad kütphanesi arduino kurulumunda yüklenmediği için sonradan kütüphane klasörüne eklenmelidir.

[Keypad kütüphane dosyası](https://drive.google.com/uc?export=download&id=17yrdcbUaFBoUbzUt7V1EAlbcBceG6c3L)nı indirmek için tıklayın…

Keypad kütüphanesini indirdikten sonra klasöre çıkarttığınız dosyayı Arduino programının kurulu olduğu klasördeki Libraries isimli klasörün içine yapıştırın.

Arduino ile keypad kullanımı için gerekli arduino kodlarını hazırlayalım.

Arduino Keypad kodları

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | #include <Keypad.h>  // keypad kütüphanesini programa dahil ediyoruz.  const byte Satir= 4;  // keypad satır sayısı  const byte Sutun= 4;  // keypad sütün sayısı  String Sifre;     // girilen şifre    // Keypad tuş yapısını hazırlıyoruz.  char key[Satir][Sutun]=  {  {'1', '2', '3', 'A'},  {'4', '5', '6', 'B'},  {'7', '8', '9', 'C'},  {'\*', '0', '#', 'D'}  };    // Arduino bağlantı bacaklarını belirliyoruz  byte SatirPinleri[Satir] = {2,3,4,5};  byte SutunPinleri[Sutun]= {6,7,8,9};    // Keypade TusTakimi ismini verip arduino bağlantılarını ve karakterleri  // bu TusTakimi keypad nesnesine tanımlıyoruz.  Keypad TusTakimi= Keypad(makeKeymap(key), SatirPinleri, SutunPinleri, Satir, Sutun);    void setup()  {    Serial.begin(9600);  }    void loop()  {    char basilanTus = TusTakimi.getKey();  // Keyped basılan tuş değerini alıyoruz.    if (basilanTus != NO\_KEY) {      Serial.print("Basılan Tuş:");      Serial.println(basilanTus);      Sifre = Sifre + basilanTus;      Serial.println("Girilen Sifre: " + Sifre );    }  } |

# **2. kaynak**

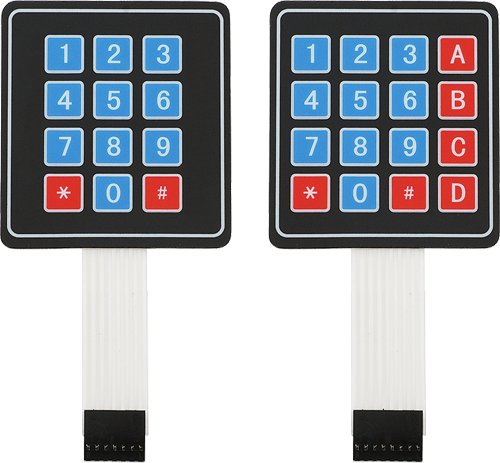
Membran matrix keypad tuş takımları cep telefonlarında, hesap makinelerinde, mikrodalga fırınlarda, kapı kilitlerinde vb. gördüğünüz tuş takımlarıdır.

Bununla birlikte, arduino projelerinde, menülerde gezinmek, şifre girmek ve robotları kontrol etmek için kullanılabilir.

**Donanıma Genel Bakış – Membran Tuş Takımı**

Membran tuş takımları ince, esnek bir plastik malzemeden yapılmıştır. 4×3, 4×4, 4×1 vb. boyutlarda olabilirler. Boyutlarından bağımsız olarak, hepsi aynı şekilde çalışır.

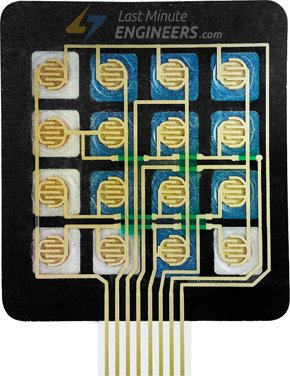
Altta membran örnek görsellerini görebilirsiniz.

****

*4×4 ve 4×3 Membran Tuş Takımı*

Örnek olarak 4×4 tuş takımını ele alalım. Toplam 16 tuşa sahiptir. Her tuşun altında özel bir membran anahtarı bulunur.

Tüm bu membran anahtarları, 4× ızgaralı bir matris oluşturan pedin altında **iletken çizgi** ile birbirine bağlanır.

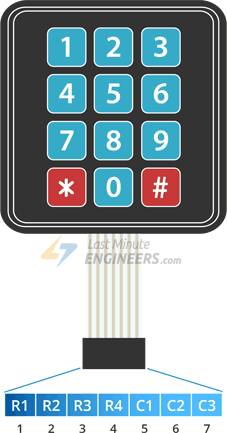
****

*4×3 Membran Tuş Takımı İç Görünümü*

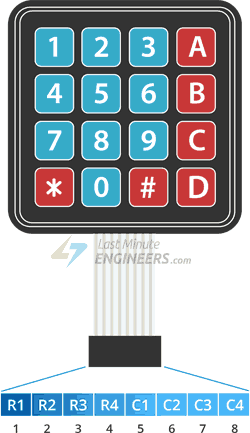
Matris mantığı olmasaydı 16 ayrı basma düğmesinin çalışması için 17 giriş pini (her tuş için bir tane ve bir toprak pini) gerekirdi. Bununla birlikte, matris düzenlemesi ile, altlığı taramak için yalnızca 8 mikrodenetleyici pini (4 sütun ve 4 sıra) yeterli olmaktadır.

**4×3 ve 4×4 Membran Tuş Takımı Pinout**

4×3 ve 4×4 membran tuş takımılarının pin çıkışları aşağıdaki gibidir.

****

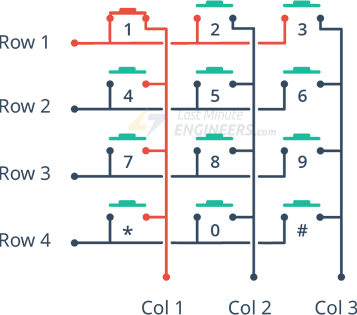
*4×3 Membran Tuş Takımı Pinout*

****

*4×4 Membran Tuş Takımı Pinout*

**Tuş Takımı Nasıl Çalışır ve Nasıl Taranır? – Matris**

Çalışma prensibi çok basittir. Bir düğmeye basılması, bulunduğu satır ve sütun üzerinde kısa devre yaparak akımlar geçmesine vesile olur. Örneğin, 4 tuşuna basıldığında, sütun 1 ve satır 2 kestirme yolu kullanılır.

****

*4×3 Membran Tuş Takımı Çalışma Prensibi*

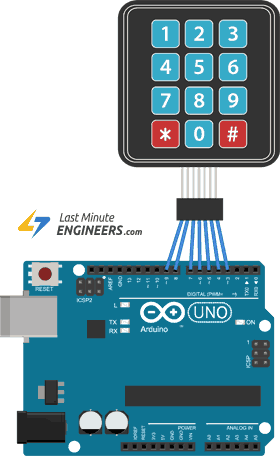
Mikrodenetleyici, basılan tuşları tarayabilir. Bunu yapmak için aşağıdaki prosedürü takip eder.

1. Mikrodenetleyici tüm sütun ve satır veri girişini algılar.
2. Ardından, bir satır seçer ve seçtiği satırı HIGH ayarlar.
3. Bundan sonra, sütun çizgilerini birer birer kontrol eder.
4. Sütun bağlantısı LOW kalırsa, satırdaki düğmeye basılmamıştır.
5. HIGH ise, mikro denetleyici hangi sıranın HIGH olarak ayarlandığını kontrol eder ve tespit eder.
6. Son olarak, algılanan satır ve sütuna karşılık gelen hangi düğmeye basıldığını bilir.

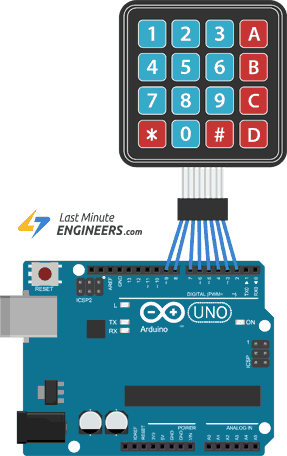
**Arduino ile 4×3 ve 4×4 Membran Tuş Takımı Bağlantısı – Kablolama**

Tuş takımı konusunda yeteri kadar bilgi sahibi olduğumuza göre arduino ile bağlamaya başlayabiliriz.

Bağlantı oldukça basittir. Tuş takımının pimini Arduino’daki dijital pim 9’a bağlayarak başlayın. Şimdi pimleri 2, 8, 3, 7 vb. gibi sola doğru bağlamaya devam edin.

****

*Arduino ile 4×3 Membran Tuş Takımı Kablolama*

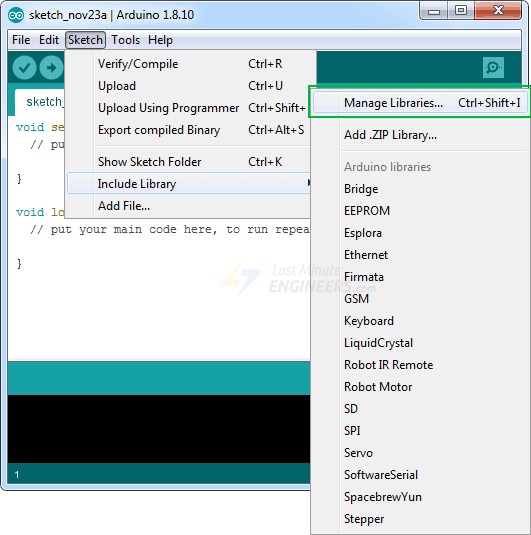
****

*Arduino ile 4×4 Membran Tuş Takımı Kablolama*

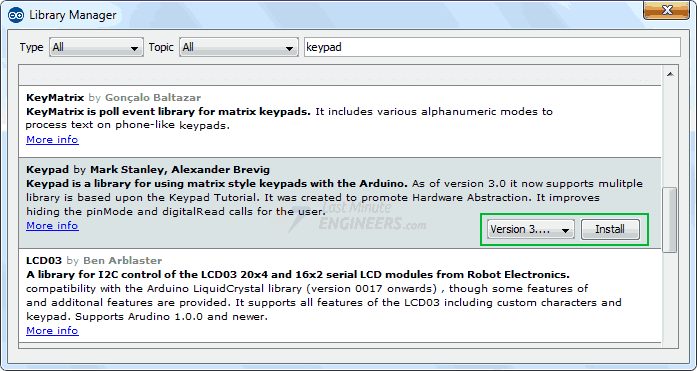
**Tuş Takımı Kitaplığını Yükleme**

Hangi tuşa basıldığını belirlemek için satırları ve sütunları sürekli olarak taramamız gerekir. Neyse ki, [Keypad.h](https://playground.arduino.cc/code/keypad" \t "_blank) bu karmaşıklığı gizlemek için yazılmıştır, böylece hangi tuşa basıldığını bilmek için basit komutlar verebiliriz.

Kitaplığı yüklemek için **Taslak> Library Ekle> Kütüphaneleri Yönet…** ‘e gidin. Kitaplık Yöneticisi’nin kitaplıklar dizinini indirmesini ve yüklü kitaplıkların listesini güncellemesini bekleyin.

****

**Keypad** yazarak aramanızı filtreleyin. Çıkan seçenekler içerisinde yaklaşık 8’inci satırda **by Mark Stanley, Alexander Brevig** olan kütüphaneyi kurun.

****

*Arduino Kütüphane Ekleme*

**Arduino Kodu**

Aşağıdaki kodlar, 4×3 ve 4×4 tuş takımı üzerinde tuşa basarak oluşturulan veri girişlerini görebileceğiniz bir proje temeli oluşturabilirsiniz.

**4×3 Tuş Takımı için kod**

#include <Keypad.h>

const byte ROWS = 4; //four rows

const byte COLS = 3; //three columns

char keys[ROWS][COLS] = {

 {'1','2','3'},

 {'4','5','6'},

 {'7','8','9'},

 {'\*','0','#'}

};

byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad

byte colPins[COLS] = {5, 4, 3}; //connect to the column pinouts of the keypad

//Create an object of keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){

**Serial**.begin(9600);

}

void loop(){

 char key = keypad.getKey();// Read the key

 // Print if key pressed

 if (key){

**Serial**.print("Key Pressed : ");

**Serial**.println(key);

 }

}

Kopyala

**4×4 Tuş Takımı Kodu**

#include <Keypad.h>

const byte ROWS = 4; //four rows

const byte COLS = 4; //four columns

char keys[ROWS][COLS] = {

 {'1','2','3','A'},

 {'4','5','6','B'},

 {'7','8','9','C'},

 {'\*','0','#','D'}

};

byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad

byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts of the keypad

//Create an object of keypad

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS );

void setup(){

**Serial**.begin(9600);

}

void loop(){

 char key = keypad.getKey();// Read the key

 // Print if key pressed

 if (key){

**Serial**.print("Key Pressed : ");

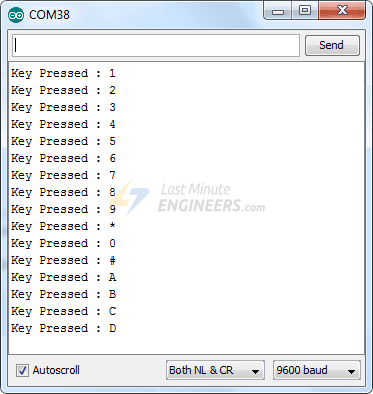
**Serial**.println(key);

 }

}

Kopyala

Ekran görüntüsü şu şekildedir.

****

*Ekran Görüntüsü*

**Kod Açıklaması**

Kod, Keypad.h kütüphanesini ekleyerek ve kullanmak istediğiniz tuş takımındaki satır ve sütun sayısını tanımlayarak başlar.

Daha sonra keys[ROWS][COLS] kodu ile tuş takımında belirli bir düğmeye basıldığında yazdırılacak karakterleri tutan 2 boyutlu bir tuş haritası dizisi tanımlarız.

Kod içerisinde karakterler, tıpkı tuş takımında göründükleri gibi düzenlenir. Aslında bunları istediğiniz herhangi bir şey olarak tanımlayabilirsiniz. Örneğin, bir hesap makinesi projesi oluşturmayı planlıyorsanız, dizi tanımını şu şekilde değiştirebilirsiniz:

char keys[ROWS][COLS] = {

 {'1','2','3','4'},

 {'5','6','7','8'},

 {'9','0','+','-'},

 {'.','\*','/','='}

};

Ardından, tuş takımı kütüphanesinin bir nesnesini oluştururuz. Keypad(makeKeymap(keys), rowPins, colPins, ROWS, COLS ) kodu 5 parametre alır.Kopyala

* makeKeymap(keys) kodu dahili tuş eşlemesini kullanıcı tanımlı tuş eşlemesine eşit olacak şekilde başlatır.
* rowPins ve colPins tuş takımının satır ve sütunlarının bağlandığı arduino pinleridir.
* ROWS ve COLS tuş takımının satır ve sütun sayısıdır.

Bir tuş takımı nesnesi oluşturulduğunda getKey() varsa hangi tuşa basıldığını kontrol etmek için basit bir komut verebiliriz.

**Tuş Takımı Kitaplığındaki Diğer Yararlı İşlevler**

**Tuş takımı** için kullanabileceğiniz birkaç kullanışlı kod vardır. Bunlardan birkaçı aşağıda listelenmiştir:

* char waitForKey() birisi bir tuşa basana kadar bekler. Yani bir tuşa basılana kadar diğer tüm kodları engeller.
* KeyState getState() tuşlardan herhangi birinin geçerli durumunu **Return** ettirir. **IDLE**, **PRESSED**, **RELEASED**, **HOLD**.
* boolean keyStateChanged() anahtarın bir durumdan diğerine ne zaman değiştiğini bilmemize yarar. Örneğin, sadece geçerli bir anahtarı test etmek yerine, bir tuşa ne zaman basıldığını test edebilirsiniz.
* setHoldTime(unsigned int time) **HOLD** durumunu değiştirecek düğme basılı tutma milisaniye miktarını ayarlar.
* setDebounceTime(unsigned int time) tuş takımının yeni bir keypress / keyEvent öğesini kabul edene kadar bekleyeceği milisaniye miktarını ayarlar.
* addEventListener(keypadEvent) hangi tuş takımı kullanılırsa o eylemi tetikler.

Hepsi bu kadar.