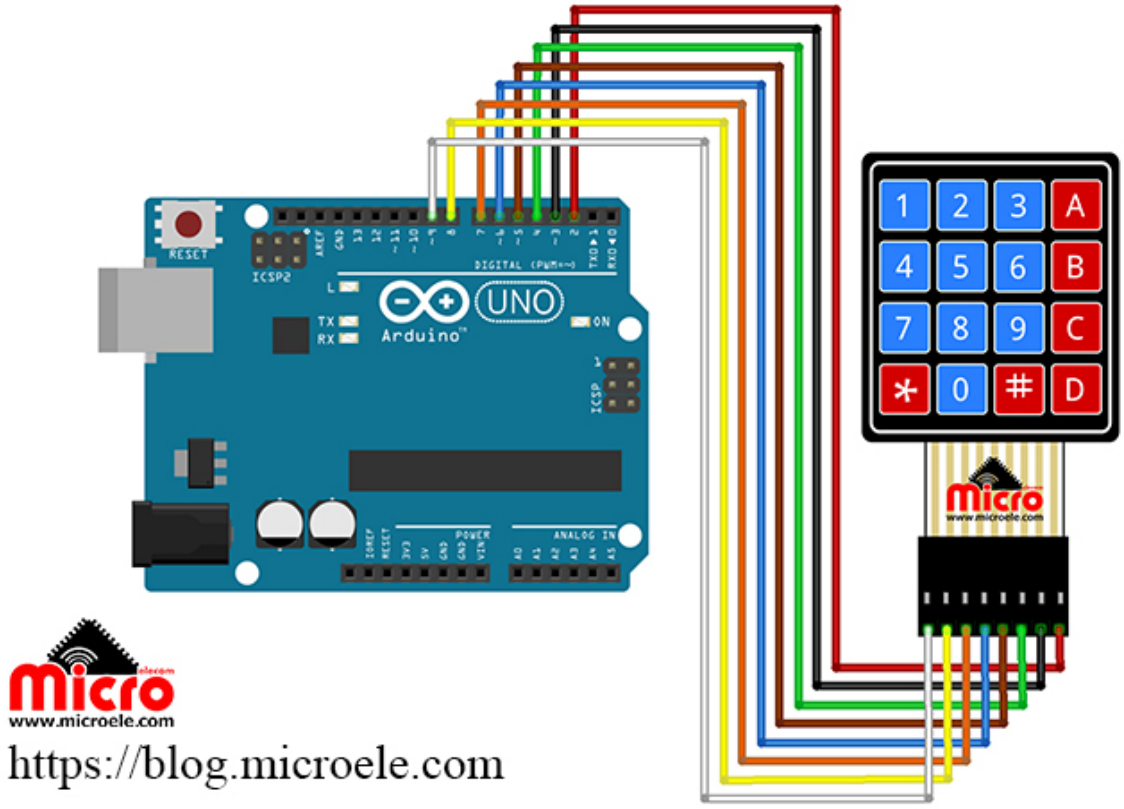
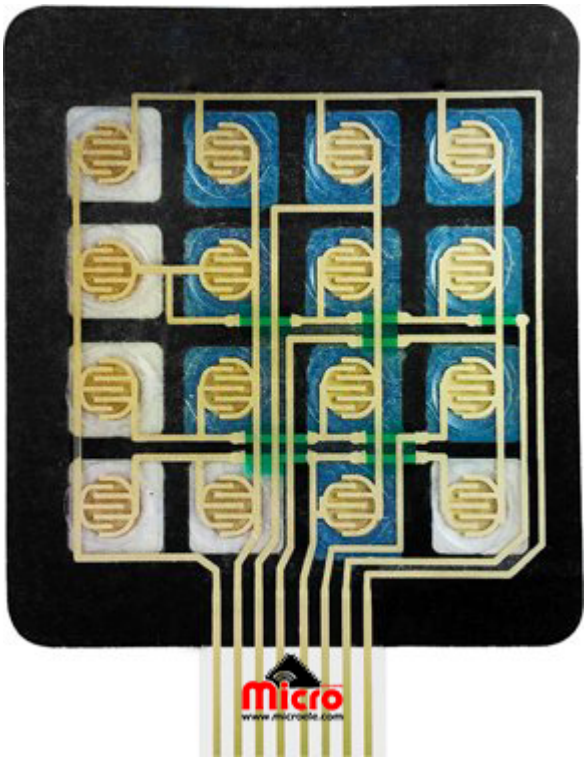
ارتباط کیپد 4×4 با آردوینو



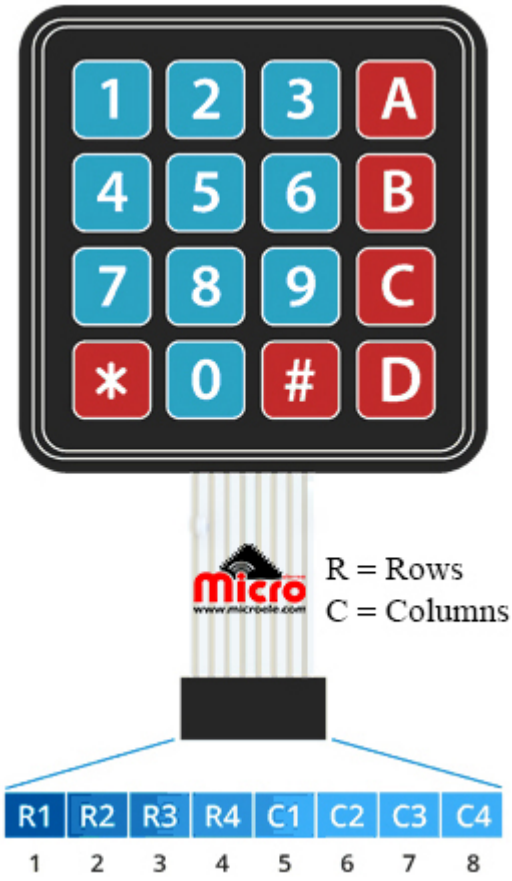
تاریخ انتشار۲۵ مرداد, ۱۳۹۹ توسط سید حسین سلطانی

صفحه کلید ها (کیپد) از تعدادی کلید فشاری تشکیل شده اند که بصورت ماتریسی به یکدیگر متصل شده اند. در اکثر پروژه ها و دستگاه های الکترونیکی مانند قفل های دیجیتالی، دستگاه های نوبت دهی، دستگاه ثبت ورود و خروج و ... از کیپد ها استفاده میشود.



امروز در این آموزش قصد داریم با هم دیگه یک کیپد 4×4 رو راه اندازی کنیم و از اون در پروژه های بعدی استفاده کنیم. پس با ما همراه باشید.

اولین قدم در راه اندازی و ارتباط کیپد به میکروکنترلر (آردوینو) شناسایی پایه های اون هست. این کیپد دارای 8 پایه هست که توسط سیم فلت در اختیار ما قرار داده شده است. 4 پایه برای سطر ها و 4 پایه دیگر هم برای ستون های کیپد.



برای خواندن اطلاعات پایه ها و این که بدونیم کاربر کدوم کلید رو فشرده است، نیاز هست که طی یک زمانی مقدار اون ها رو بررسی کنیم. مثلا به سطر 1 منطق یا ولتاژ '0' رو اعمال کنیم و پایه های ستون رو بررسی کنیم ببینیم کدوم صفر هست مقدارش.

بهنوان مثال اگر ستون 2 مقدارش '0' بود یعنی کاربر عدد 3 رو انتخاب کرده. با این حساب میتونیم بفهمیم کدوم کلید فشرده شده و کاربر چه عددی رو انتخاب کرده. و همین روند رو برای بقیه سطر ها و ستون ها تکرار کنیم.

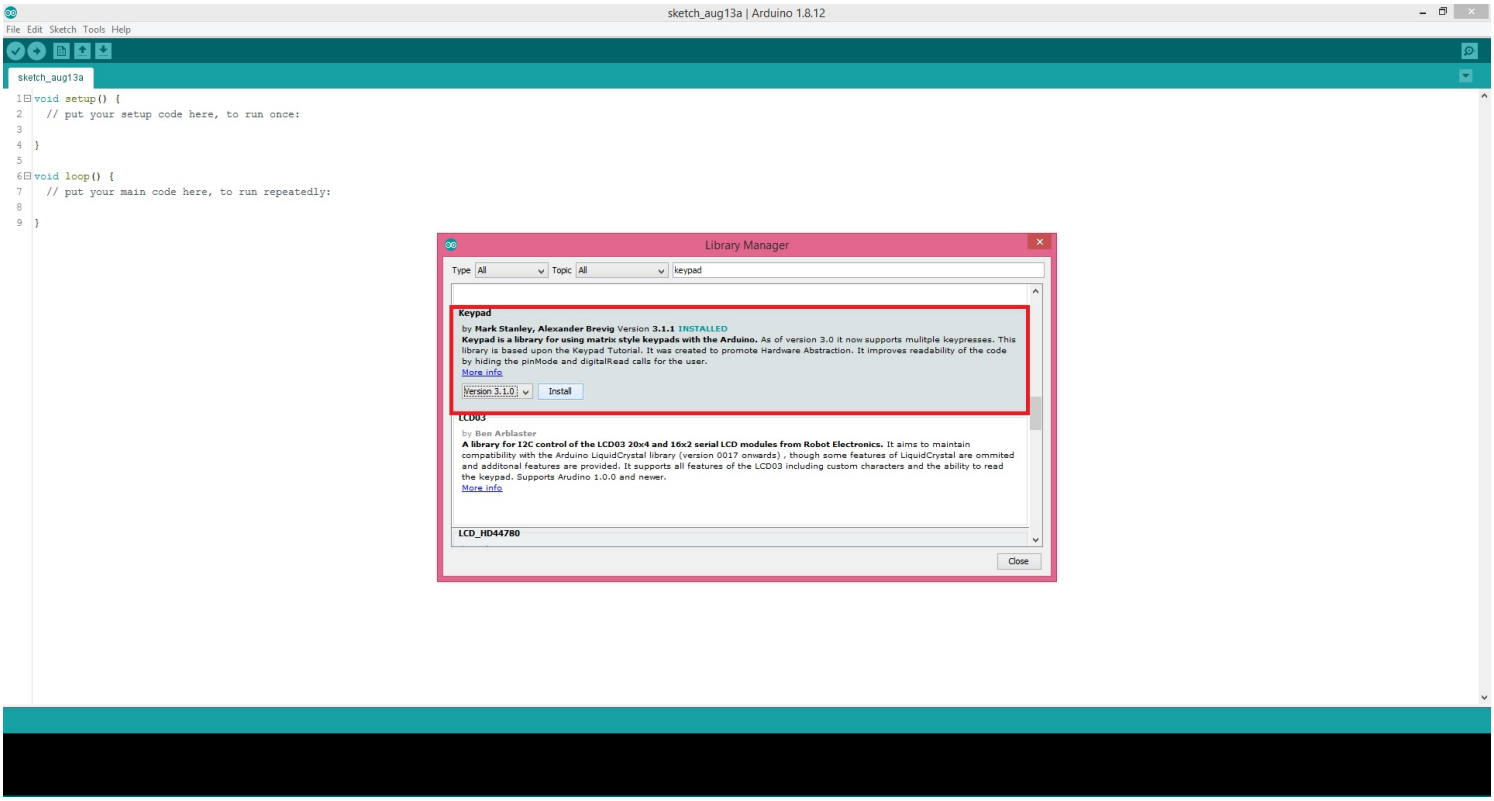
خب حالا بریم سراغ یه روش دیگه که کار برنامه نویسی رو کمتر میکنه و در عین حال راحت تر به جواب مطلوبمون خواهیم رسید. از اونجا که آردوینو یک پلتفرم متن باز هست، کتابخانه ای برای ارتباط کیپد وجود داره که از اون استفاده خواهیم کرد.

اضافه کردن کتابخانه Keypad:

برای اضافه کردن این کتابخانه به نرم افزار ARDUINO IDE مراحل زیر رو دنبال کنید:

Sketch >> Include Library >> Manage Library

بعد از انتخاب گزینه Manage Library، پنجره ای به شکل زیر نمایش داده خواهد شد. در قسمت مشخص شده عبارت keypad رو جست و جو کرده و کتابخانه مشخص شده در کادر زیر رو نصب می کنیم.



در زیر به چند بخش از این تابع اشاره خواهیم کرد.

فراخوانی دستورات Keypad:

در ابتدا تعداد سطر و ستون را مشخص میکنیم:

```
const byte Rows = 4; //4 سطر
const byte Columns = 4; //4 ستون
```

در یک آرایه 2 بعدی اعداد و کاراکتر ها را مطابق با کیپد مشخص میکنیم:

```
char keys[Rows][Columns] =
  {
    { '1', '2', '3', 'A', },
    { '4', '5', '6', 'B', },
    { '7', '8', '9', 'c', },
    { '*', '0', '#', 'D'  }
  };
```

در 2 آرایه دیگر شماره پین هایی که کیپد به آردوینو وصل میشود را مشخص میکنیم:

```
byte rowPins[Rows] = { 9, 8, 7, 6 }; // پین های متصل به سطر
byte colPins[Columns] = { 5, 4, 3, 2 }; // پین های متصل به ستون
```

در این قسمت باید یک آبجکت از کلاس keypad.h تعریف کنیم تا بتوانیم عدد انتخاب شده از کیپد را مورد بررسی قرار دهیم:

```
Keypad Key_pad= Keypad (makeKeymap (keys), rowPins, colPins, Rows, Columns);
```

این کتابخانه دارای توابع کاربردی می باشد که در زیر به توابع پر کاربرد آن اشاره خواهیم کرد:

تابع getKey():

با استفاده از این تابع، کاراکتر مربوط به کلید فشرده شده را بدست خواهیم آورد. در این تابع در صورتی که کلیدی فشرده نشود باعث ایجاد اختلال در روند برنامه یا مسدود شدن در آن نمی شود.

```
void loop ()
{
  char getKey = Key_pad.getKey();
  if (getKey)
  {
    Serial.print(getKey);
  }
}
```

تابع Waitforkey:

همان طور که از این تابع مشخص است، منتظر میماند تا کلیدی فشرده شود. طبعا بخاطر این منتظر بودن باعث متوقف شدن اجرای کد های دیگر غیر از وقفه ها خواهد شد.

```
void loop ()
{
  char getKey = Key_pad.waitForKey();
  if (getKey)
  {
    Serial.print(getKey);
  }
  Serial.print("Hosein Soltani");
}
```

تابع getkeys():

با کمک این تابع می توانیم از فشرده شدن یک کلید مطلع شویم. در صورتی که کلیدی فشرده شد مقدار True و در غیر این صورت مقدار False را بر میگرداند.

```
void loop()
{
  if ( Key_pad.getKeys() )
  {

    Serial.print("Pressed ...");
  }
}
```

تابع addEventListener(KetpadEvent):

با استفاده از این تابع یک رخداد (event) کیپد به برنامه اضافه خواهد شد. برای استفاده از این حالت یک تابع دلخواه در حلقه loop معرفی میشود.

```
void loop()
{
  char getKey = Key_pad.getKey ();
}

void keypadEvent ( KeypadEvent key )
```

```
{

}
```

در مرحله دوم باید در تابع تنظیمات یا setup به وسیله دستور addEventListener تابعی که در بالا نوشته شده است را به برنامه اصلی معرفی کنیم.

```
void setup()
{
  Serial.begin ( 9600 );
  Key_pad.addEventListener( keypadEvent );
}
void loop()
{
  char getKey = Key_pad.getKey ();
}
void keypadEvent ( KeypadEvent key )
{
  کد های مد نظر برای اجرا شدن بعد از ورود به این تابع//
}
```

این تابع دارای توابع وابسته دیگه ای هم هست. چون حجم مطالب زیاد میشه و باهاشون فعلا کاری نداریم از گفتنشون خودداری میکنیم.

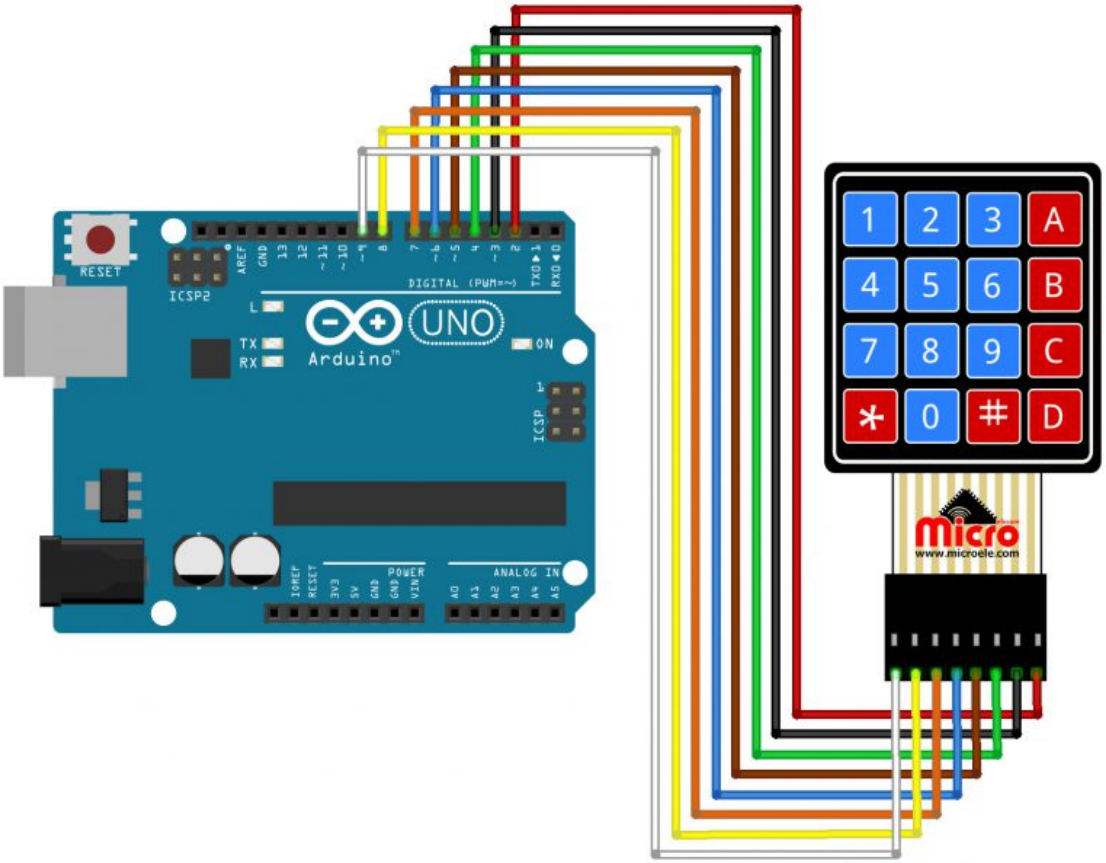
وسایل مورد نیاز:

برای استفاده از این آموزش نیاز به اقلام زیر داریم:

- **آردوینو UNO**

- کیپد 4×4
- سیم برد بوردی

شماتیک:

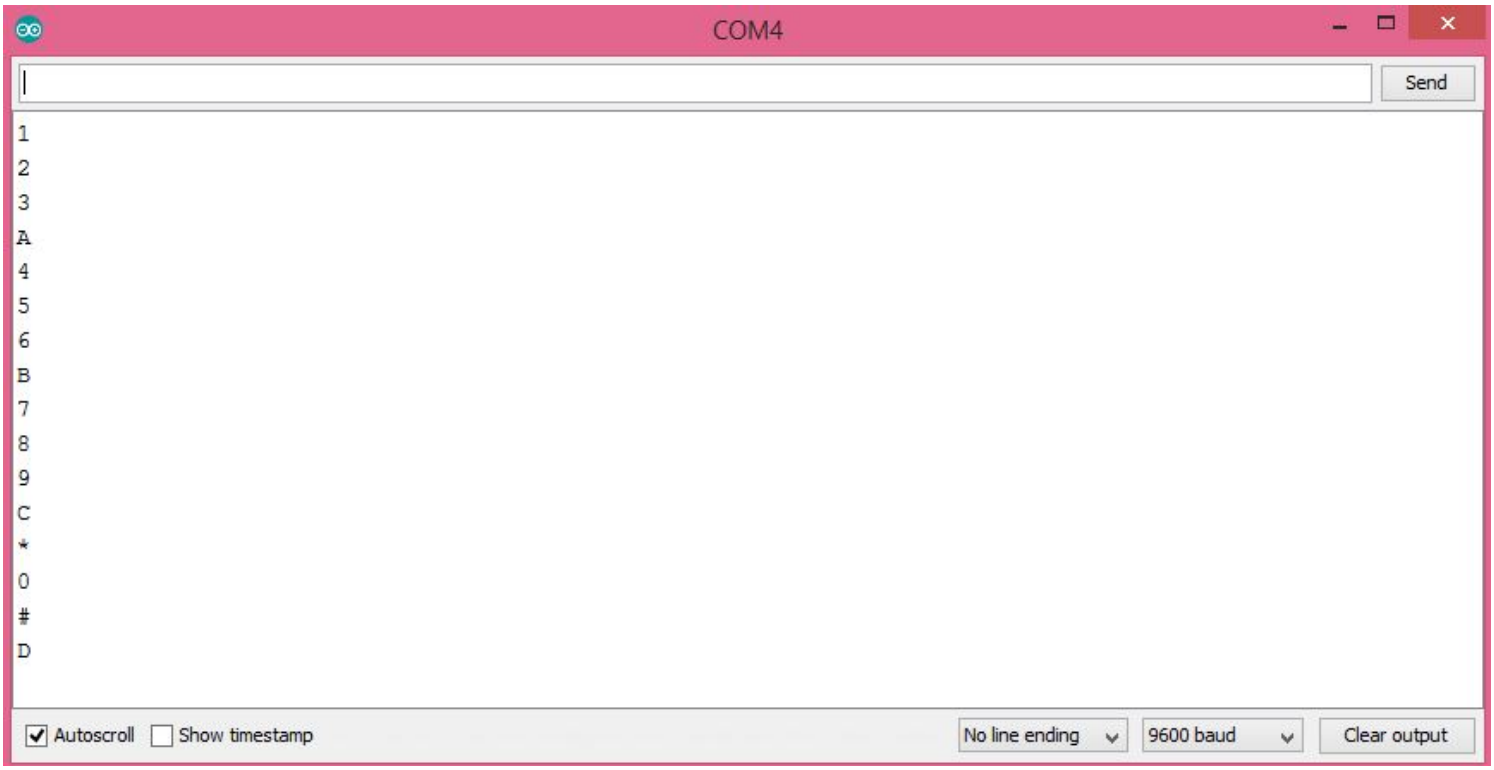


کد های برنامه:

برای برنامه نویسی و نمایش عدد انتخاب شده کیپد، یک پروژه جدید ایجاد کنید. و کد های زیر را در محیط برنامه نویسی وارد کنید. برای سادگی و جلوگیری از سیم کشی زیاد، به جای نمایش کاراکتر ها روی یک LCD از سریال

مانیتور نرم افزار استفاده میکنیم و در اون کاراکتر ها رو نمایش خواهیم داد.

```
#include <Keypad.h>;
const byte Rows = 4;
const byte Columns = 4;

char keys[Rows][Columns] = {
    { '1', '2', '3', 'A' },
    { '4', '5', '6', 'B' },
    { '7', '8', '9', 'C' },
    { '*', '0', '#', 'D' }
  };
byte rowPins[Rows] = { 9, 8, 7, 6 }; // پین های متصل به سطر
byte colPins[Columns] = { 5, 4, 3, 2 }; // پین های متصل به ستون

Keypad Key_pad = Keypad (makeKeymap (keys), rowPins, colPins, Rows, Columns);

void setup ()
  {
    Serial.begin (9600);
  }

void loop()
{
  char key = Key_pad.getKey();

  if (key)
  {
    Serial.println(key);
  }
}
```



در آخر این نکته رو اشاره کنم که هدف از این مطلب آموزش راه اندازی کیپد بود. اگر پروژه یا آموزشی در این خصوص دارید اون را با ما به اشتراک بگذارید تا با نام خودتون داخل سایت انتشار پیدا کنه.

در صورت هرگونه سوال یا پیشنهادی اون رو کامنت کنید و در اولین فرصت خیلی سریع بهش پاسخ میدم. همچنین اگر از این آموزش استفاده کردید و پروژه رو انجام دادید اون رو با ما در اینستاگرام با #microelecom به اشتراک بگذارید.

برای دانلود کد های این آموزش می توانید از این لینک استفاده کنید.