

شروع کار با نرم افزار آردوینو



این مطلب دومین قسمت از **آموزش جامع آردوینو (مبتدی و پیشرفته)** است. تا اینجا سخت افزار آردوینو را شناختید و دیدید که با آن چه کارهای جالبی می‌توانید انجام دهید. زمان آن رسیده که آردوینوی دلخواهتان را انتخاب کنید و شروع به برنامه‌نویسی برای آن کنید. اولین قدم برای شروع کار با آردوینو یادگیری کار با نرم افزار آن است. در این آموزش با نحوه نصب نرم افزار و بخش‌های مختلف آن آشنا خواهید شد و چند دستور ساده آنرا نیز با هم تمرین خواهیم کرد. در پایان این آموزش می‌توانید یک پروژه ساده آردوینو را خودتان اجرا کنید.

راه ارتباطی ما با برد آردوینو، نرم‌افزاری است به نام **Arduino IDE** که تمام برنامه‌ها درون آن نوشته می‌شود. اگر بخواهیم به طور خلاصه فرایندی که برای کار با آردوینو طی می‌شود را توضیح دهیم، به این صورت است: آردوینو را از طریق **USB** به رایانه متصل می‌کنید؛ برنامه مورد نظرتان را در **IDE** می‌نویسید؛ برنامه را به روی برد می‌فرستید؛ از آن به بعد آردوینو همواره برنامه شما را به صورت خط به خط از بالا به پایین اجرا خواهد کرد، تا اینکه برنامه دیگری بر روی آن بریزید یا اینکه برق آنرا قطع کنید.

نصب نرم افزار آردوینو در ویندوز

برای شروع باید نرم افزار **IDE آردوینو** را داشته باشید. می‌توانید با مراجعه به **سایت آردوینو** نرم افزار را دانلود کنید. خوشبختانه نرم افزار برای سیستم عامل‌های مختلف موجود است. برای ویندوز می‌توانید نسخه نصبی و یا نسخه فشرده را دانلود کنید. پیشنهاد می‌شود از نسخه نصبی استفاده کنید تا همزمان، درایورهای **USB** هم نصب شود. در صورتی که نسخه فشرده را دانلود کنید، باید درایور مربوطه را نیز به صورت دستی نصب کنید.

Download the Arduino IDE

ARDUINO 1.8.8
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the Getting Started page for installation instructions.

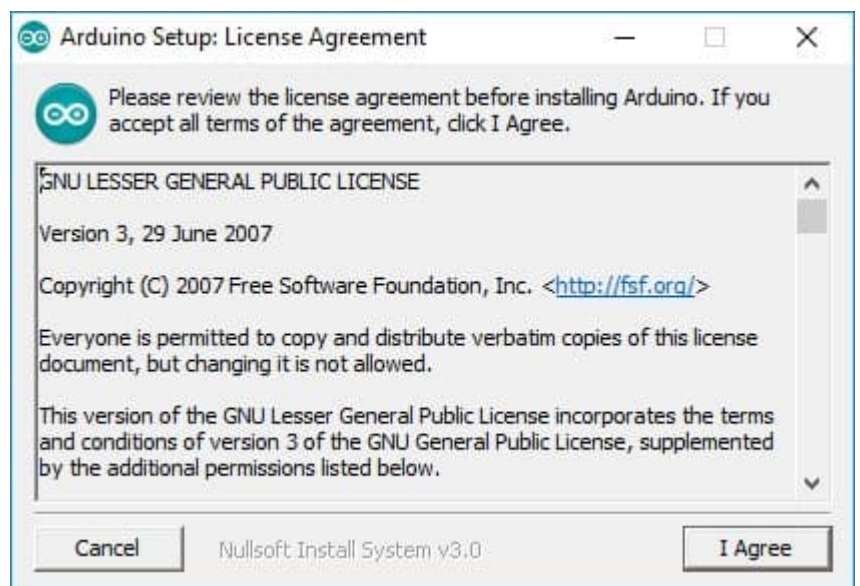
Windows installer, for Windows XP and up
Windows ZIP file for non admin install
Windows app Requires Win 8.1 or 10
Get

Mac OS X 10.8 Mountain Lion or newer

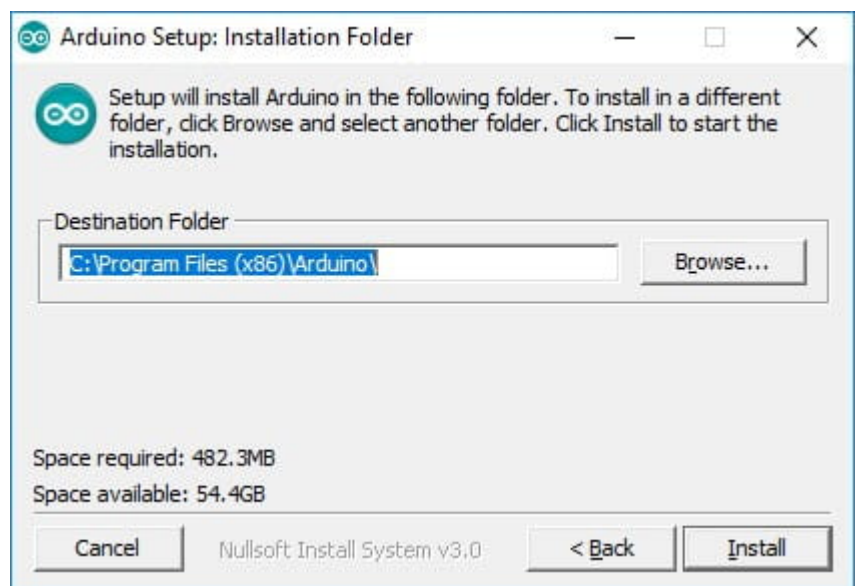
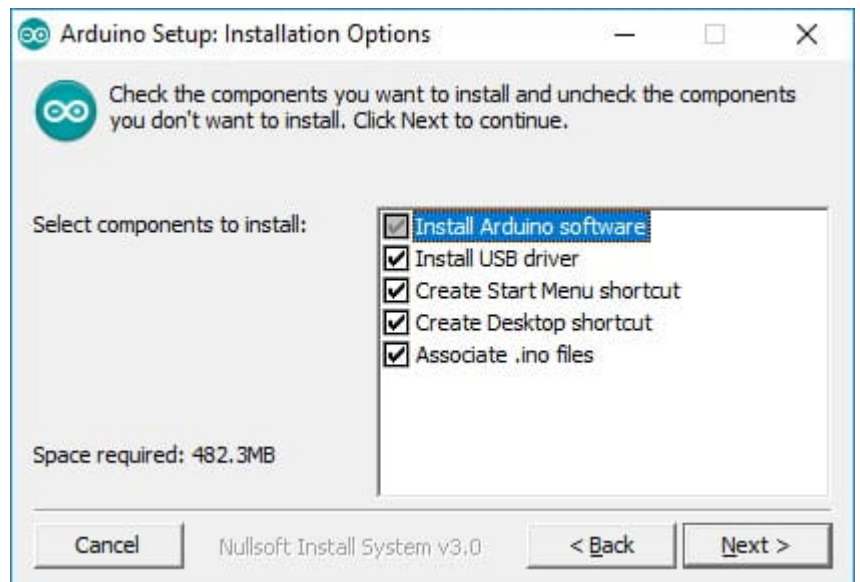
Linux 32 bits
Linux 64 bits
Linux ARM

Release Notes
Source Code
Checksums (sha512)

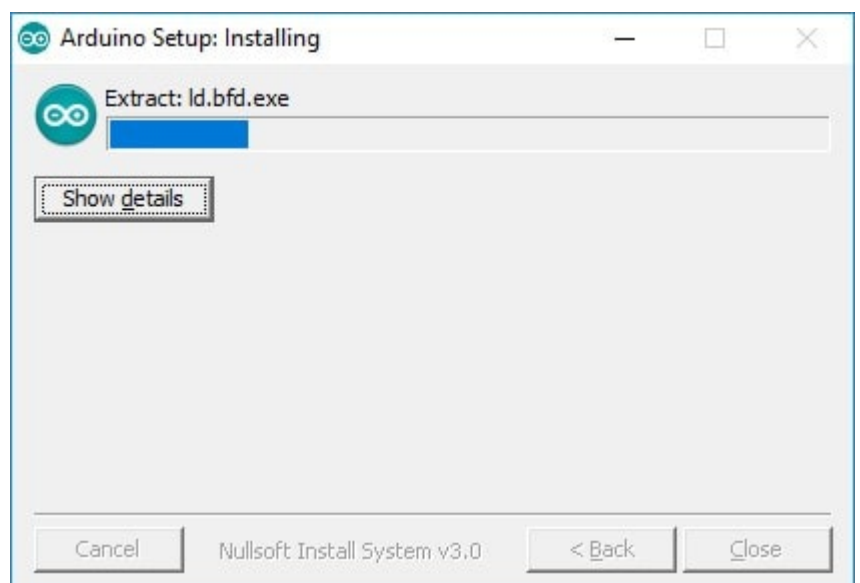
پس از اینکه برنامه را دانلود کردید، آنرا اجرا کنید. اگر پیغام امنیتی از طرف سیستم عامل دریافت کردید، به برنامه اجازه دسترسی دهید. شرایط نصب را تایید کنید.



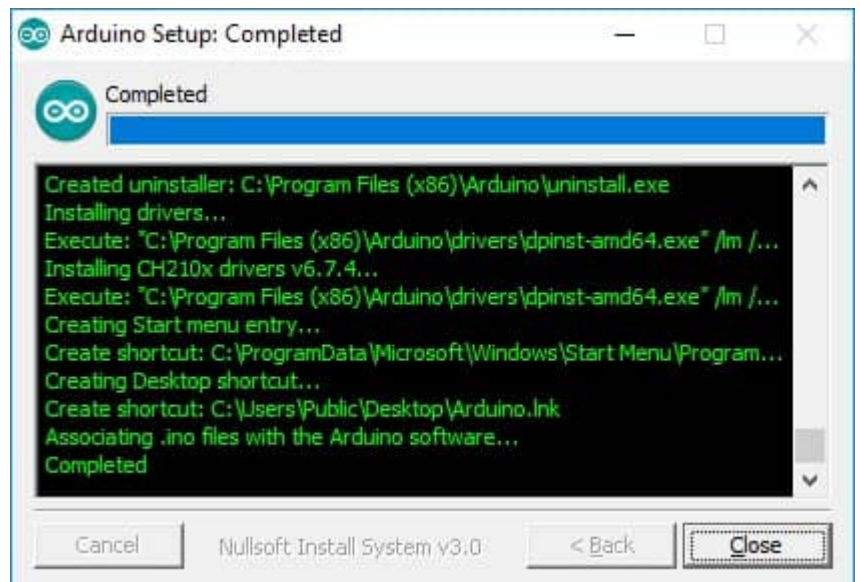
می‌توانید اجزای برنامه جهت نصب و محل نصب را تعیین کنید اما پیشنهاد می‌کنیم برنامه را با تنظیمات پیش‌فرض نصب کنید.



منتظر بمانید تا نصب نرم افزار انجام شود.

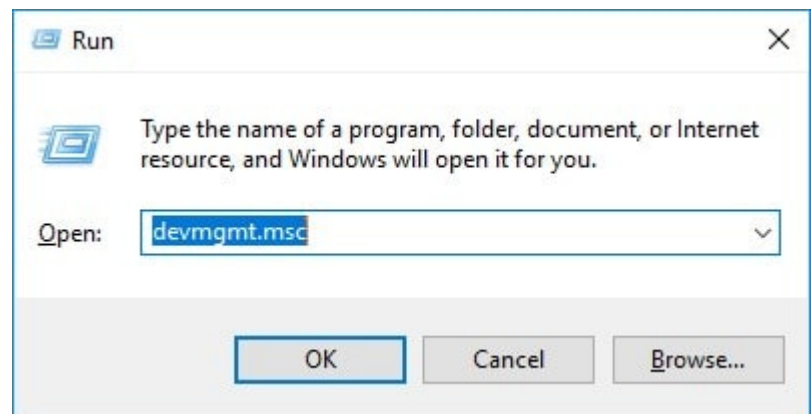


در نهایت پیغام اتمام نصب نمایش داده می شود. نصب نرم افزار تمام است.

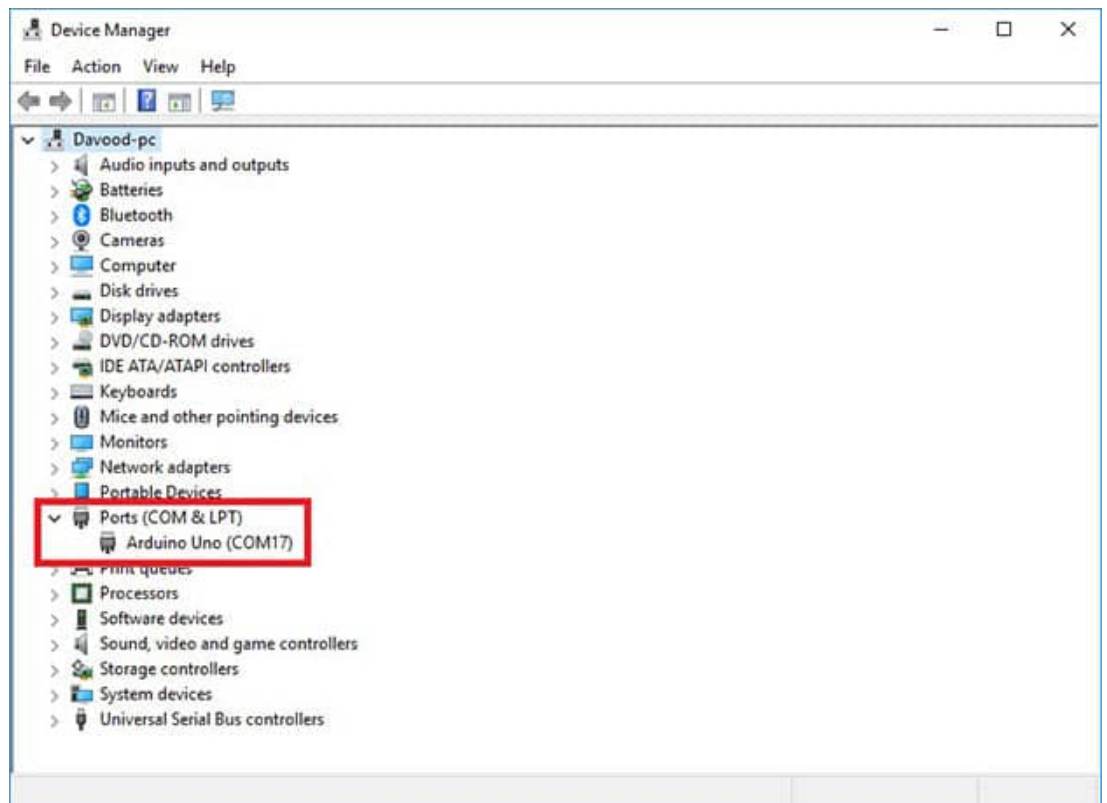


اتصال برد آردوینو به رایانه

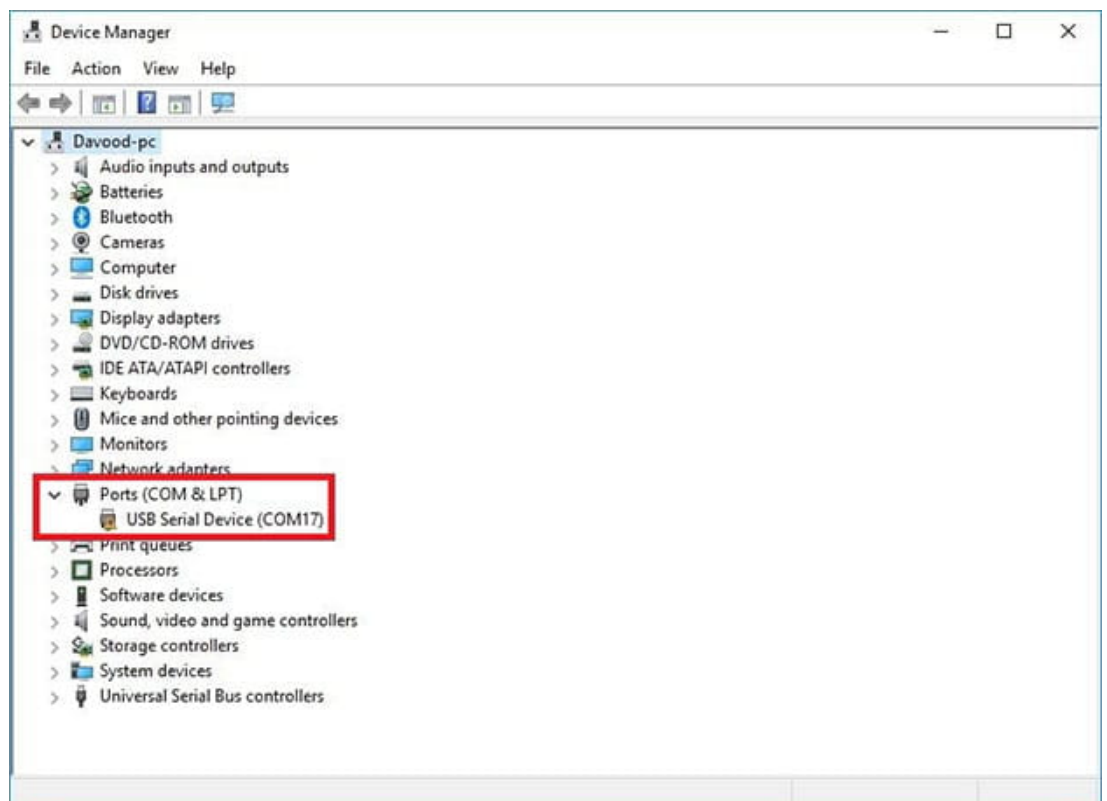
آردوینو را به پورت USB رایانه وصل کنید و منتظر بمانید تا شناسایی شود. برای اطمینان از اینکه برد آردوینو به درستی شناسایی شده است، کلید **ctrl+R** را فشار دهید و در پنجره جدید عبارت `devmgmt.msc` را تایپ کرده و **Ok** را بزنید.



پنجره مشخصات سخت‌افزاری سیستم باز می‌شود. در لیست قطعات، گزینه **Ports** را باز کنید. در صورتی که آردوینو به صورت عادی در آن وجود داشت، سیستم شما مشکلی در اتصال با برد آردوینو ندارد.

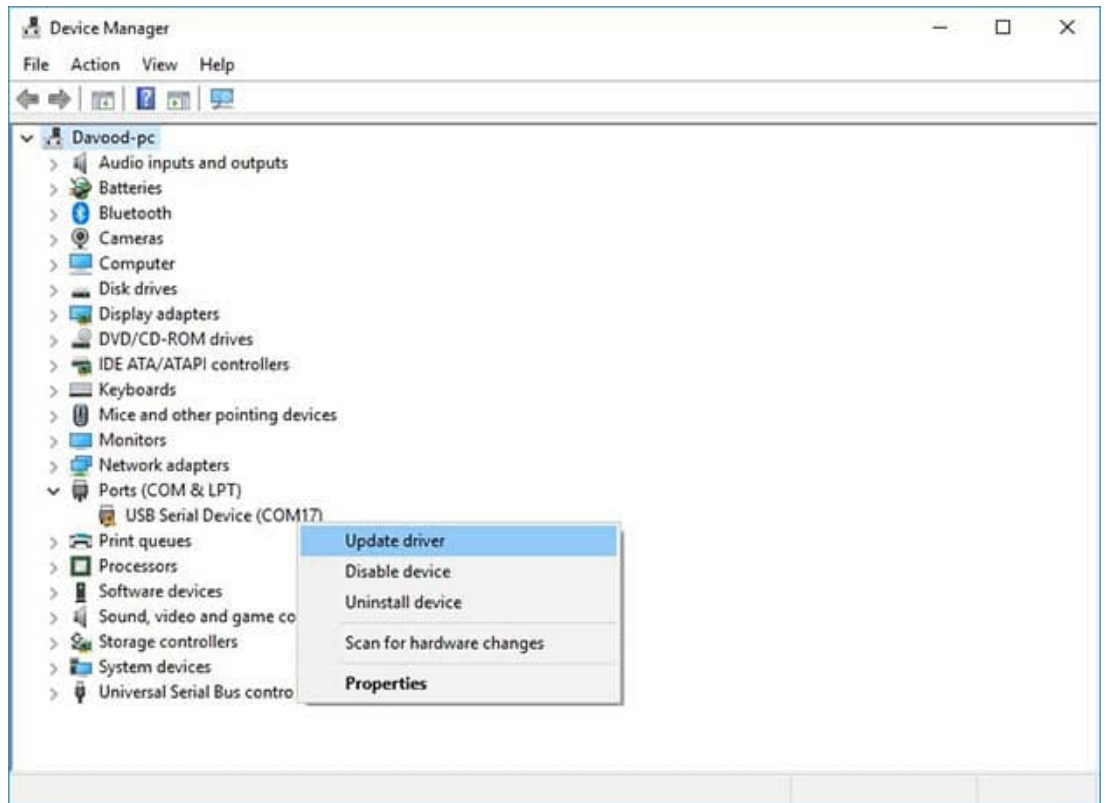


در صورتی که بر روی آیکون آن علامت زرد رنگ باشد یا نام مدل آردوینو نوشته نشده باشد، به این معنی است که احتمالاً درایور آردوینو به درستی نصب نشده است.

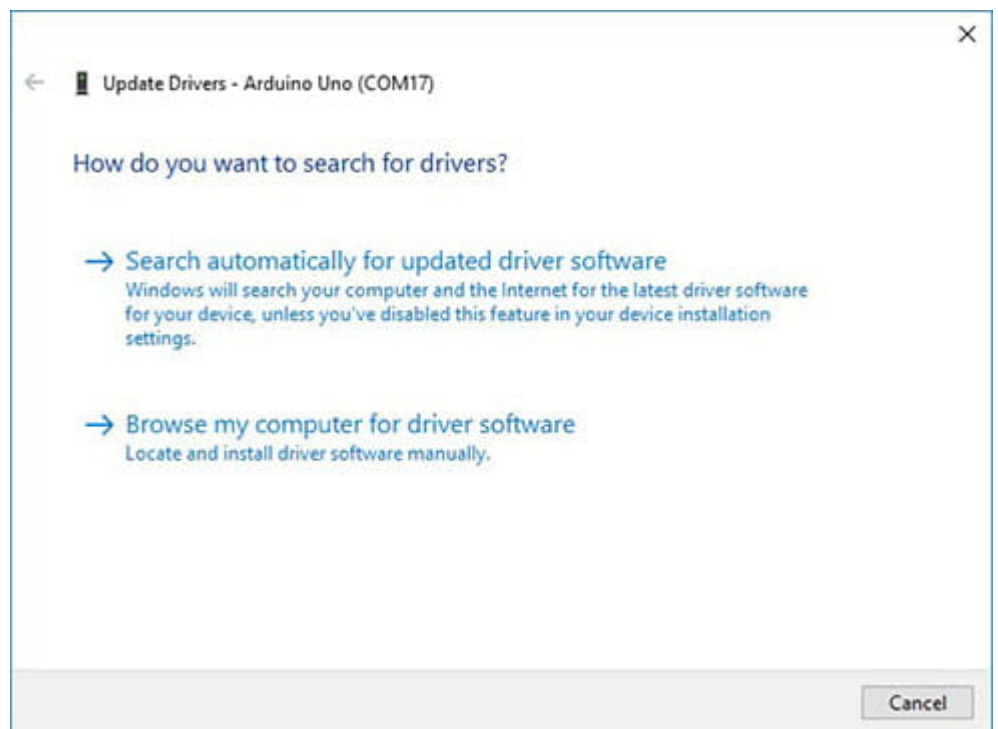


نصب درایور آردوینو

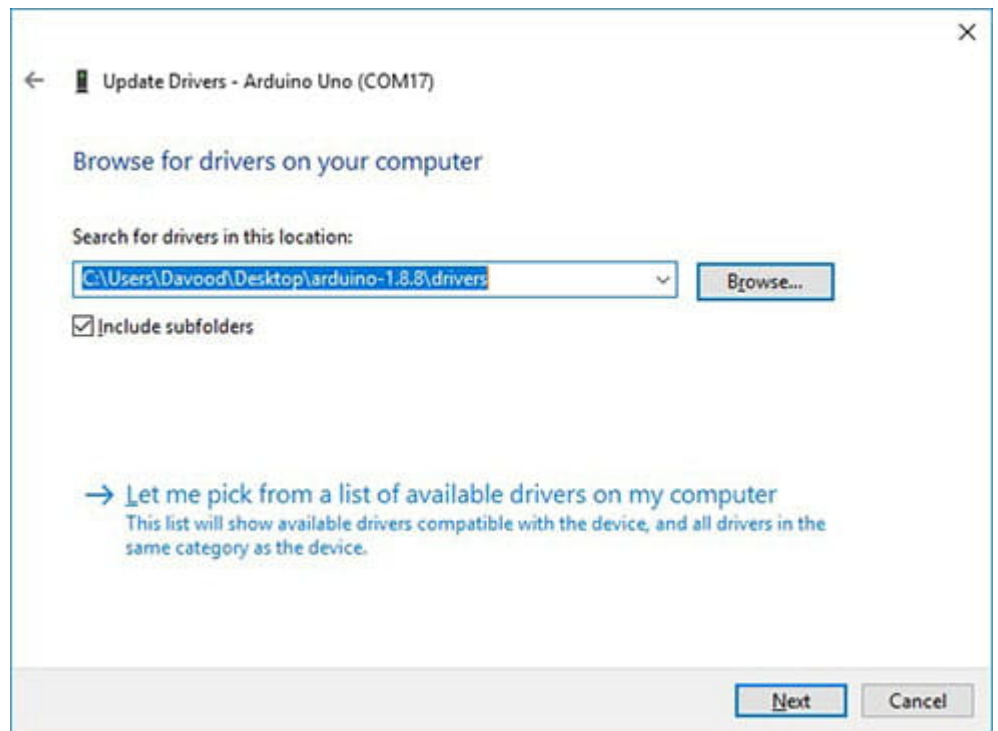
اگر قبلاً نسخه فشرده نرم‌افزار را دانلود نکرده‌اید، این کار را انجام دهید. برای نصب درایور در همان پنجره‌ی مشخصات سخت‌افزاری، بر روی پورت مربوط به آردوینو راست کلیک کرده و گزینه Update driver را انتخاب کنید.



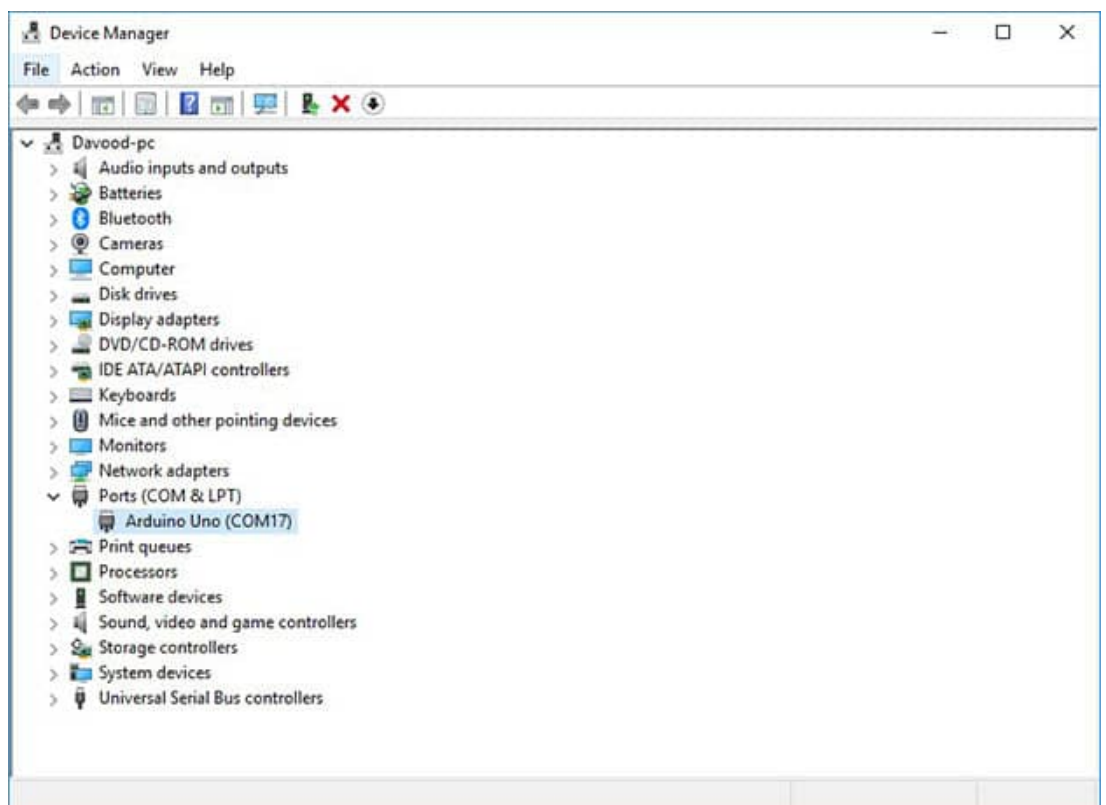
در پنجره جدید گزینه دوم (Browse my computer for driver software) را انتخاب کنید.



فایل فشرده‌ای که قبلاً دانلود کرده‌اید را از حالت فشرده خارج کنید. حالا آدرس پوشه فوق را در پنجره آپدیت درایور وارد کرده و تیک Include subfolders را بزنید. کلید Next را بزنید و منتظر بمانید تا نصب درایور تمام شود.



اگر نصب به درستی صورت گرفته باشد، در پنجره مشخصات سخت‌افزاری، باید آیکون آردوینو در قسمت Ports به حالت عادی در آمده باشد.



نصب نرم‌افزار آردوینو در لینوکس و مک

نصب برنامه آردوینو در لینوکس و MacOS به مراتب ساده‌تر است. برای این کار در MacOS، ابتدا نرم‌افزار را از سایت آردوینو دانلود کنید، آنرا از حالت فشرده خارج کرده و سپس آن را به فولدر Applications منتقل کنید. آردوینو را به رایانه متصل کنید. در پنجره‌ای که باز می‌شود، بر روی Network preferences و سپس بر روی Apply کلیک کنید. پنجره را ببندید. به فولدر Applications رفته و آردوینو را بر روی Dock بکشید و رها کنید.

برای نصب آردوینو در لینوکس روش‌های مختلفی موجود است. ابتدا نسخه نرم‌افزار را با توجه به نوع لینوکس نصب شده، از سایت دانلود کنید. Terminal را باز کنید و

به محل دانلود بروید:

```
cd ~/Downloads
```

فایل دانلود شده را از حالت فشرده خارج کنید (بجای fileName نام فایلی که دانلود کرده‌اید را بنویسید):

```
tar -xvf fileName
```

محتویات پوشه نصبی را به مسیر opt منتقل کنید:

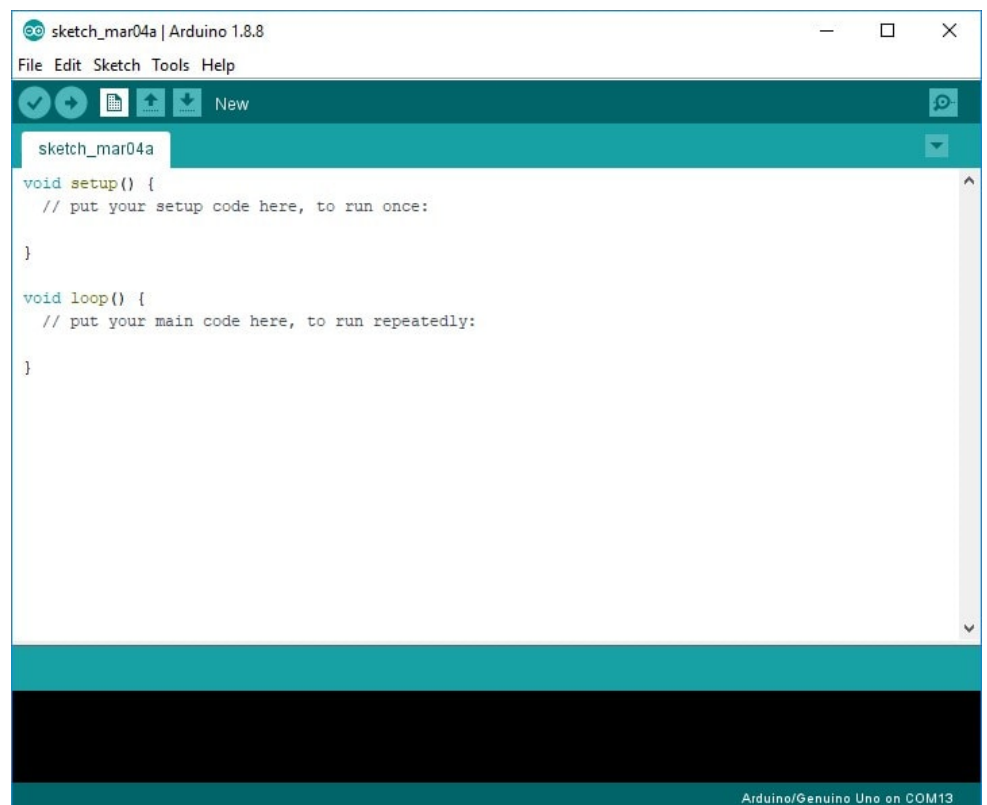
```
sudo mv fileName /opt
```

به پوشه مورد نظر بروید و دستورات زیر را اجرا کنید:

```
cd /opt/fileName/  
chmod +x install.sh  
./install.sh
```

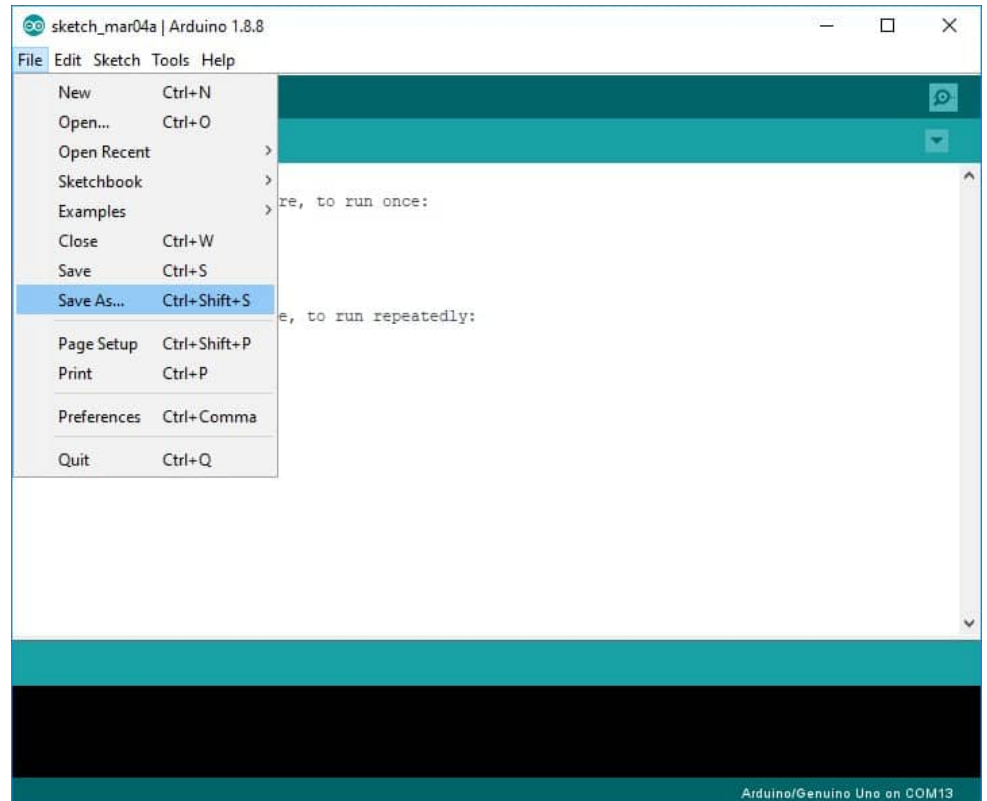
ساختار برنامه آردوینو

نرم‌افزار آردوینو نصب شد؛ به همین راحتی! همه چیز در آردوینو به همین سادگی است. بریم به سراغ معرفی بخش‌های مختلف نرم‌افزار. تا اینجا نرم‌افزار آردوینو را نصب کرده و اتصال برد به رایانه را انجام داده‌اید. وقتی نرم‌افزار را برای اولین بار باز می‌کنید یک کد خام آردوینو را برای شما نمایش می‌دهد.



خود نرم‌افزار فایل را به صورت پیش‌فرض بر اساس تاریخ میلادی نام‌گذاری می‌کند. اگر می‌خواهید یک پروژه را شروع کنید، بهتر است که ابتدا یک اسم برای برنامه‌اش

انتخاب کنید. برای تغییر نام فایل می‌توانید از مسیر **File** > **Save As** این کار را انجام دهید.



حلقه‌های اصلی

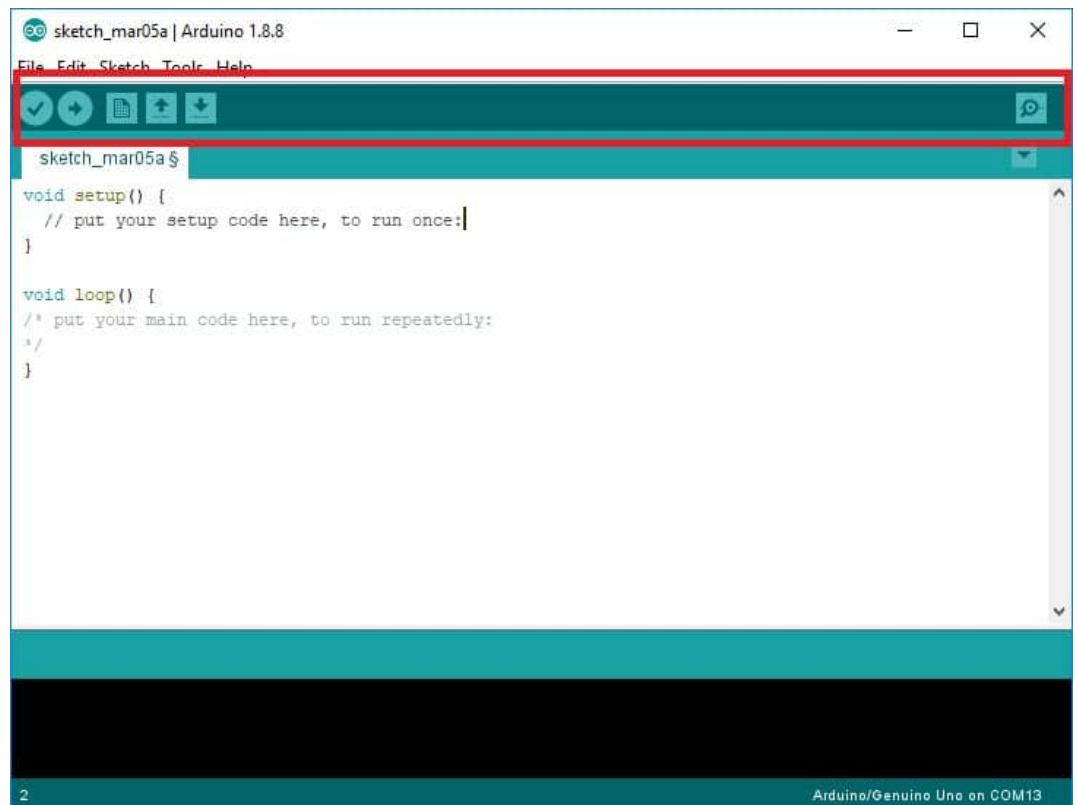
در آردوینو به هر برنامه، یک **Sketch** یا طرح گفته می‌شود. مبنای کدنویسی آردوینو زبان **C** و **C++** است که برای راحتی کار با آن، تا حد زیادی ساده‌سازی شده است. یک برنامه آردوینو حداقل شامل دو بخش کلی است: **void setup** و **void loop**. در ادامه‌ی هر کدام از این بخش‌ها دو آکولاد وجود دارد. محتوای هر کدام از این بخش‌ها بین این دو آکولاد تعریف می‌شود. **setup** مجموعه‌ای از کدها را در بر می‌گیرد که هر وقت آردوینو به برق متصل شود یا کلید **Reset** فشرده شود، تنها یک بار اجرا می‌شود. **loop** مجموعه‌ای از کدها را در خود دارد که باید بدون وقفه و به صورت تکراری اجرا شوند.

```
void setup() {
  //This code will be executed once.
}

void loop() {
  //This code will be executed repeatedly.
}
```

توضیحات (کامنت)

در نرم‌افزار این قابلیت وجود دارد که در بین متن کد، توضیحات بنویسید. این توضیحات توسط کامپایلر نرم‌افزار پردازش نمی‌شود و فقط مربوط به خواننده کد است. این توضیحات کمک می‌کند تا اگر فرد دیگری کدتان را بخواند یا خودتان بعد از مدتی دوباره از آن استفاده کردید، بتوانید آن را متوجه شوید. با گذاشتن دو اسلش، // آن خط به حالت کامنت در می‌آید و می‌توانید در آنجا توضیحات خودتان را بنویسید. این نوع کامنت فقط یک خط بوده و با رفتن به خط بعد برنامه به حالت عادی برمی‌گردد. علامت **/*** نیز همین کار را می‌کند با این تفاوت که کامنت گذاری تا جایی که علامت ***/** را بگذارید ادامه پیدا می‌کند. به بیان دیگر هر چیزی که بین دو علامت **/*** و ***/** قرار بگیرد به معنای کامنت خواهد بود و این نوع توضیحات می‌تواند چند خط باشد. این دو نوع کامنت را می‌توانید در تصویر زیر مشاهده کنید.

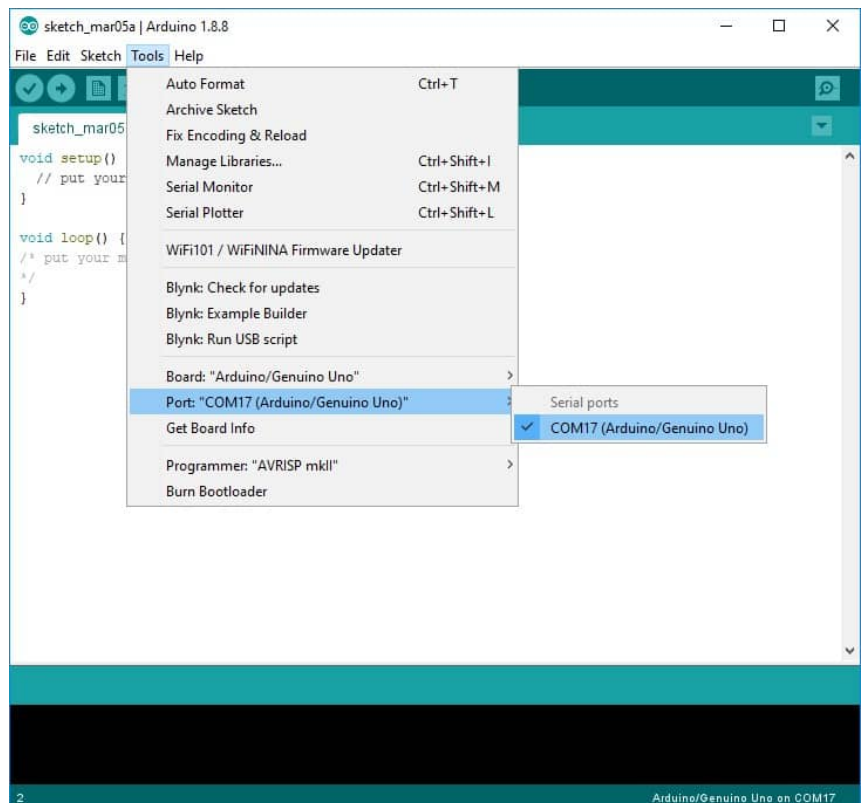
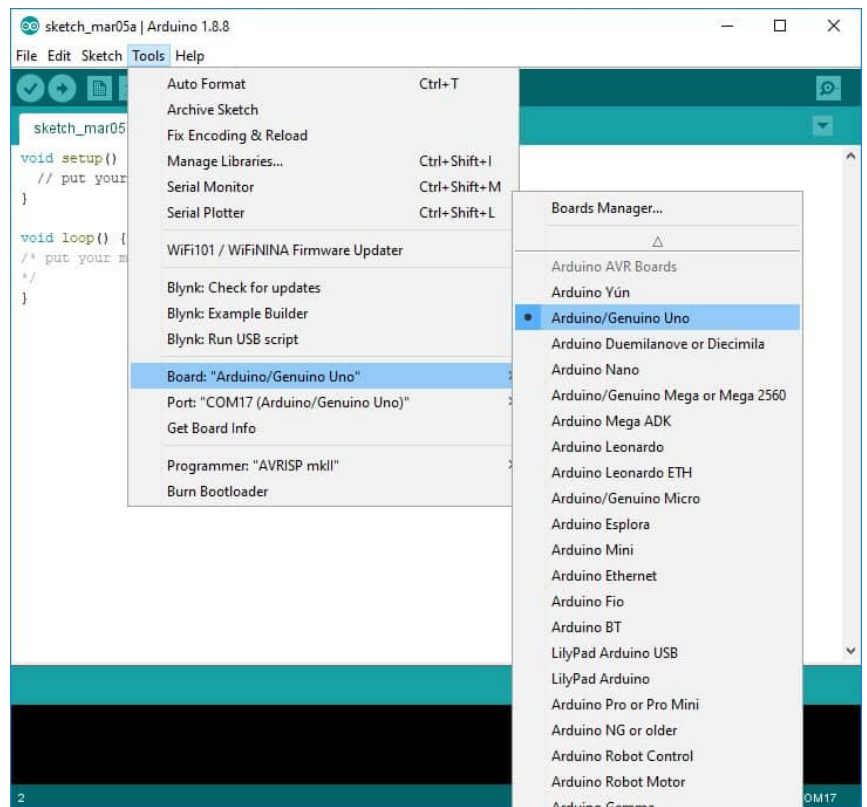


نوار ابزار

در نوار بالایی نرم‌افزار تعدادی کلید مشاهده می‌کنید که برای دسترسی سریعتر قرار داده شده‌اند. از سمت چپ، کلید اول Verify است. این کلید بررسی می‌کند که آیا کد شما مشکلی برای اجرا دارد یا نه، اما آنرا روی برد نمی‌ریزد. این قابلیت زمانی که بردی به سیستم متصل نیست یا به هر دلیلی نمی‌خواهید کد به برد انتقال داده‌شود، به کارتان می‌آید. کلید دوم یا Upload ابتدا کد را کامپایل کرده. کلید سوم یا New یک پنجره جدید باز می‌کند. کلید چهارم یا Open برای باز کردن برنامه‌هایی که از قبل نوشته‌اید و کلید پنجم یا Save هم برای ذخیره کردن برنامه به کار می‌رود.

انتخاب پورت و برد

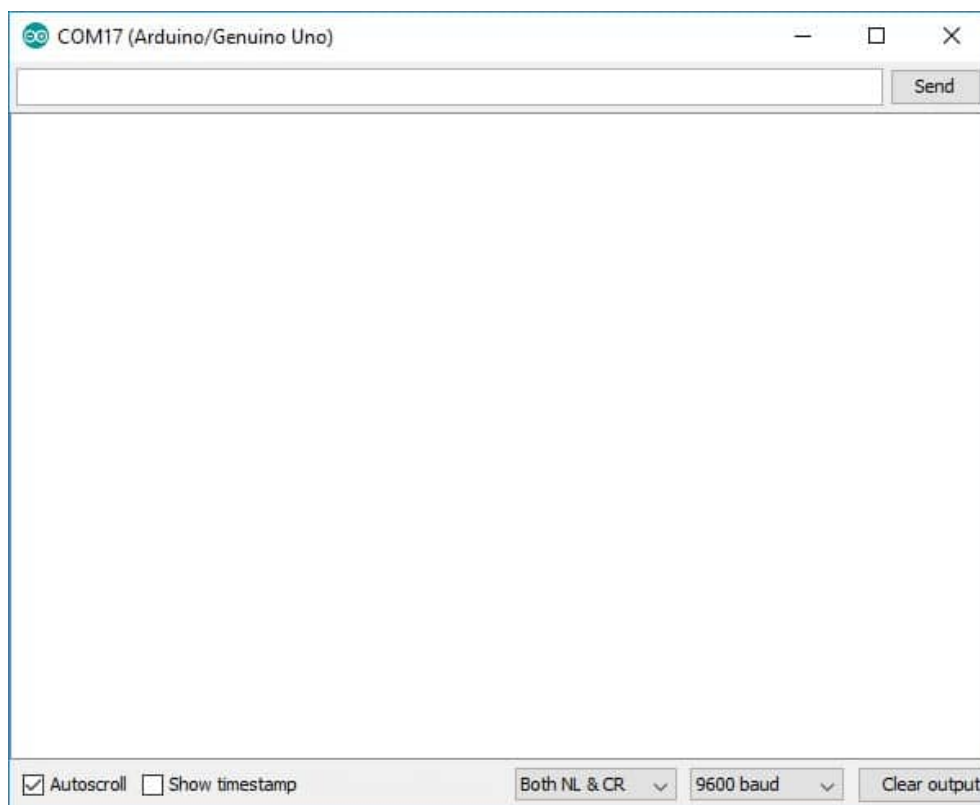
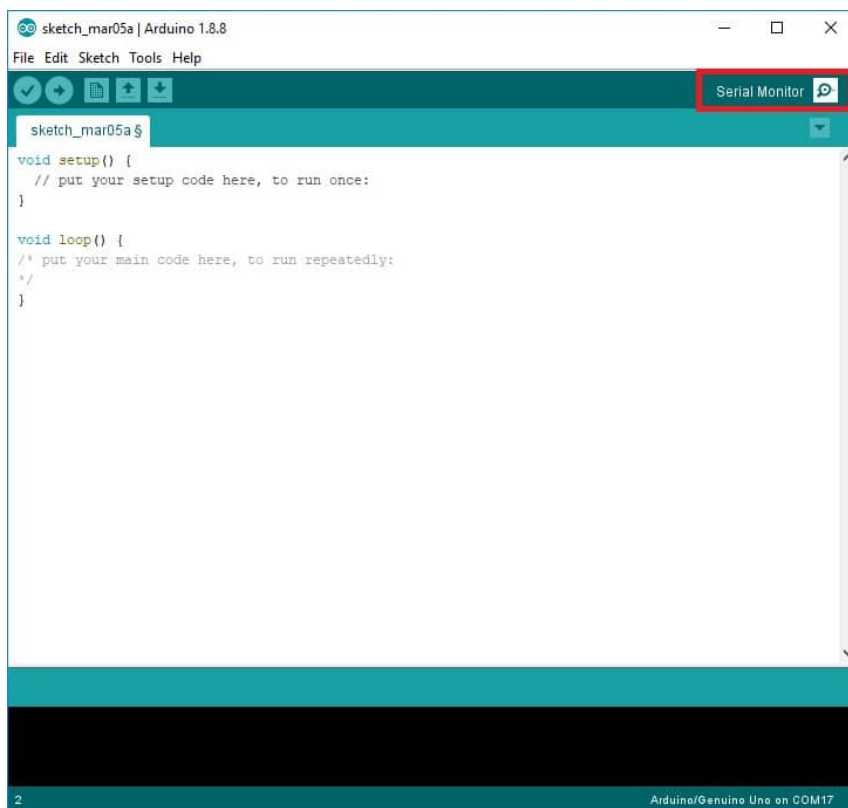
هر زمان که یک برد آردوینو را به رایانه متصل می‌کنید، قبل از هر چیز باید برد و پورت مورد استفاده را به برنامه معرفی کنید. برای این منظور از مسیر **Tools** > **Board** مدل و از مسیر **Tools** > **Port** پورتهای که آردوینو به آن متصل شده است را انتخاب کنید.



نام پورت اتصال ممکن است هر دفعه با دفعه پیش متفاوت باشد، پس همیشه از صحت آن مطمئن شوید. برای این کار می‌توانید از قسمت اتصال برد آردوینو به رایانه که در بالا توضیح داده شد، نام پورت را پیدا کنید.

سریال مانیتور

در آردوینو محیطی به نام سریال مانیتور (Serial Monitor) وجود دارد که در آن می‌توانید متن مورد نظرتان را چاپ کنید. کلیدی که در انتهای سمت راست نوار بالا وجود دارد پنجره سریال مانیتور را برای شما باز می‌کند.



در سریال مانیتور می‌توانید خروجی یک تابع، اطلاعات یک مازول یا هر اطلاعات دیگری بنویسید. این کار به شما کمک می‌کند تا از روند اجرای برنامه مطلع شوید و

بتوانید آنرا عیب‌یابی کنید. علاوه بر آن در نوار بالای سریال مانیتور می‌توانید یک دستور یا عبارت را به برد ارسال کنید که در بخش‌های آینده در این مورد توضیح خواهیم داد.

برای استفاده از سریال مانیتور کافیسیت در حلقه `setup` دستور زیر را وارد کنید:

```
Serial.begin(9600)
```

عدد درون پرانتز، سرعت داده‌برداری سریال را نشان می‌دهد و می‌توانید مقدار آنرا عوض کنید.

کتابخانه در آردوینو

فرض کنید که می‌خواهید یک ماژول که از پروتکل سریال استفاده می‌کند را راه‌اندازی کنید. برای برقراری ارتباط با این پروتکل نیاز به کمی کدزنی دارید تا توابع مورد نظران را تعریف کنید. اگر درون یک برنامه بخواهید چند کار مختلف انجام دهید یا با ماژول‌های مختلف کار کنید، تعداد خطوط کد مربوط به تعاریف می‌تواند آنقدر زیاد شود که کد اصلی در آن گم شود. از این بدتر هم می‌تواند پیش بیاید: اگر بخواهید بعداً در یک برنامه دیگر همان ماژول را استفاده کنید، باید کدهای مربوطه را پیدا کرده و در برنامه جدید کپی کنید. این مجموعه کدها معمولاً پرکاربرد هستند و می‌توانند آنقدر کاربردی باشند، که لازم داشته باشید در پروژه‌های مختلفی از آن استفاده کنید. راه حل چیست؟ چطور می‌شود از این همه کد تکراری خلاص شد؟

در برنامه‌نویسی آردوینو، قابلیت وجود دارد که این مسئله را به کلی حل کرده‌است. کتابخانه‌ها مجموعه‌ای از تعاریف و توابع هستند که یک بار برای همیشه نوشته می‌شوند و از آن پس، فقط مورد استفاده قرار می‌گیرند. استفاده از کتابخانه، حجم کدتان را کم کرده، برنامه‌تان خواناتر و قابل فهم‌تر شده و منابع سخت‌افزاری کمتری استفاده می‌کند. استفاده از یک کتابخانه بسیار ساده است. کافی است در ابتدای برنامه دستور زیر را وارد کنید (بجای عبارت `Library` نام کتابخانه مورد نظران را بنویسید).

```
#include <Library.h>
```

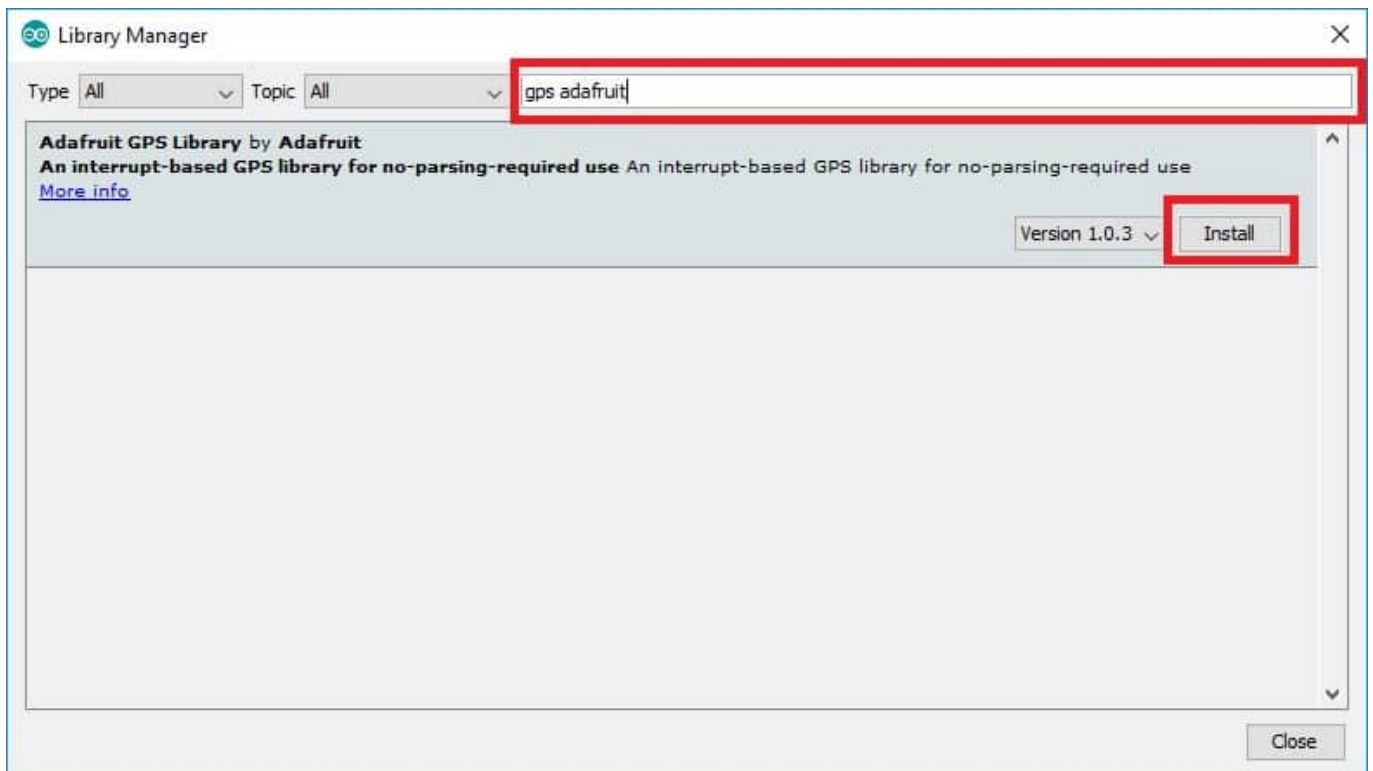
با این کار تمام قابلیت‌های کتابخانه مورد نظران را در اختیار خواهید داشت. برای مثال دستور زیر کتابخانه مربوط به ارتباط نمایشگر کاراکتری را فراخوانی می‌کند:

```
#include <LiquidCrystal.h>
```

برای استفاده از یک کتابخانه، ابتدا باید آنرا به نرم‌افزار اضافه کنید. برای این منظور چند روش وجود دارد.

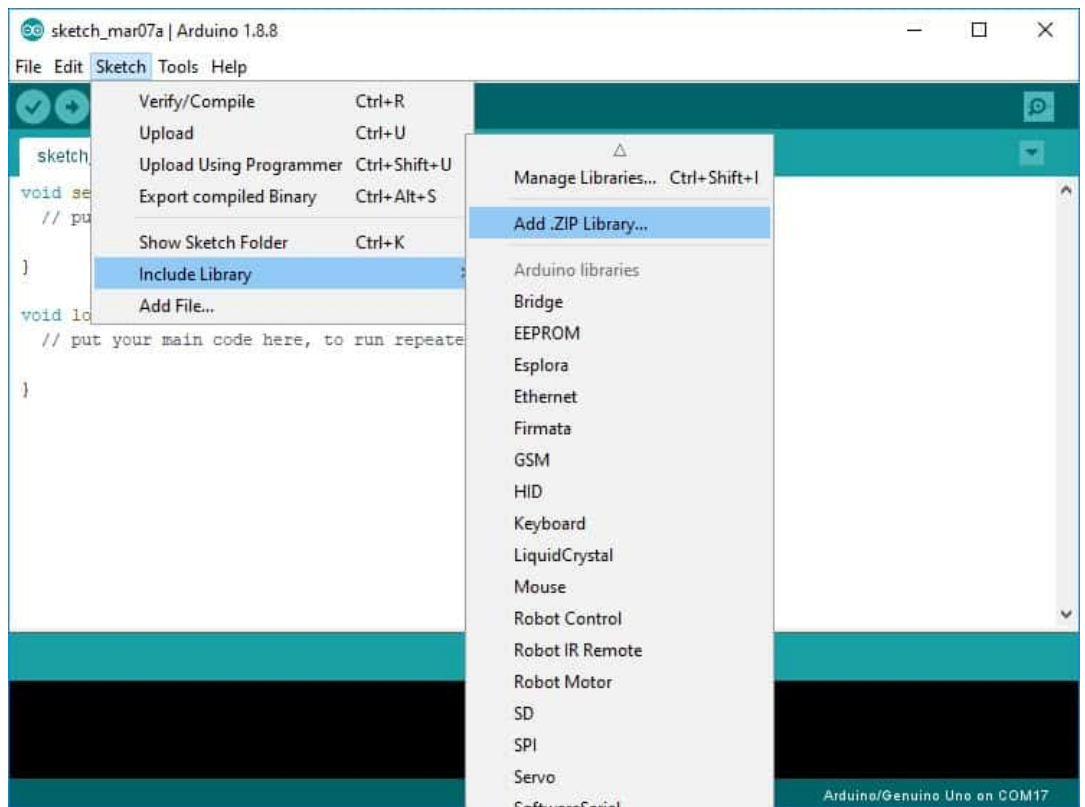
روش اول

ساده‌ترین راه برای نصب یک کتابخانه، دانلود با خود نرم‌افزار است. برای این کار به مسیر `Sketch > Include Library > Manage Libraries` بروید. چند لحظه صبر کنید تا لیست کتابخانه‌ها فراخوانی شود. در این پنجره می‌توانید لیستی از کتابخانه‌های آردوینو را ببینید. در نوار سمت راست می‌توانید عنوان کتابخانه را جستجو کرده و با کلیک بر روی گزینه `Install` کتابخانه مورد نظر را نصب کنید.



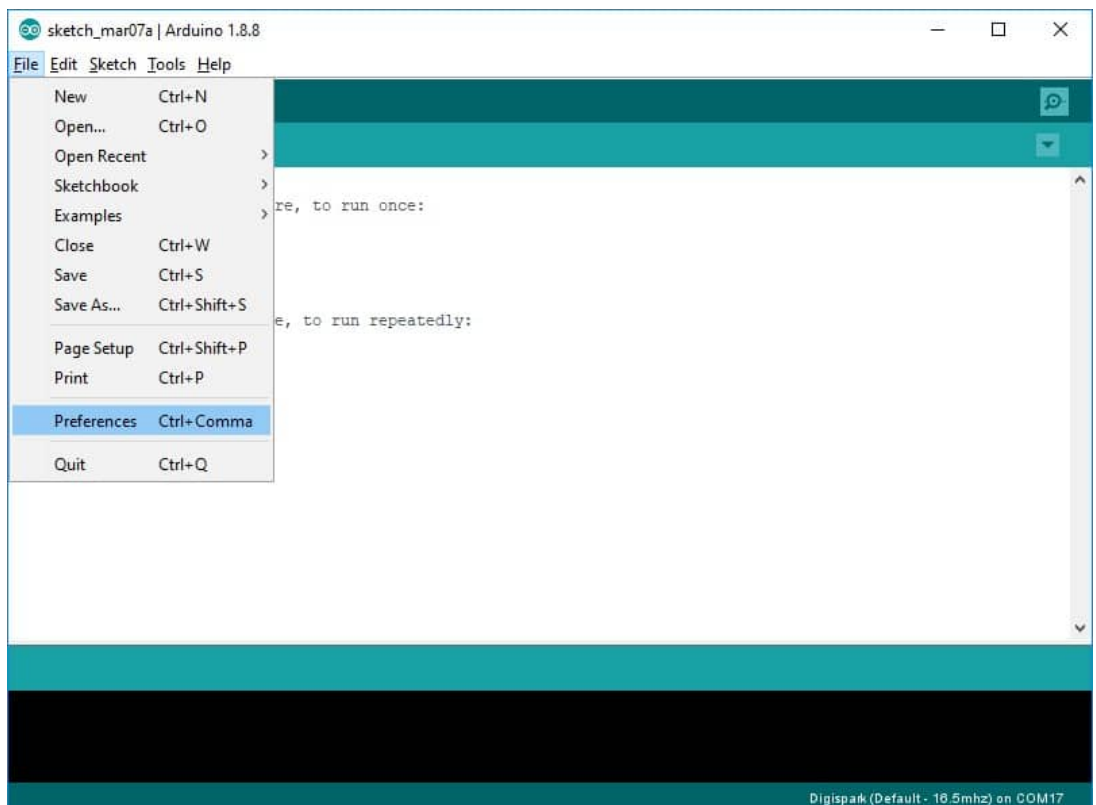
روش دوم

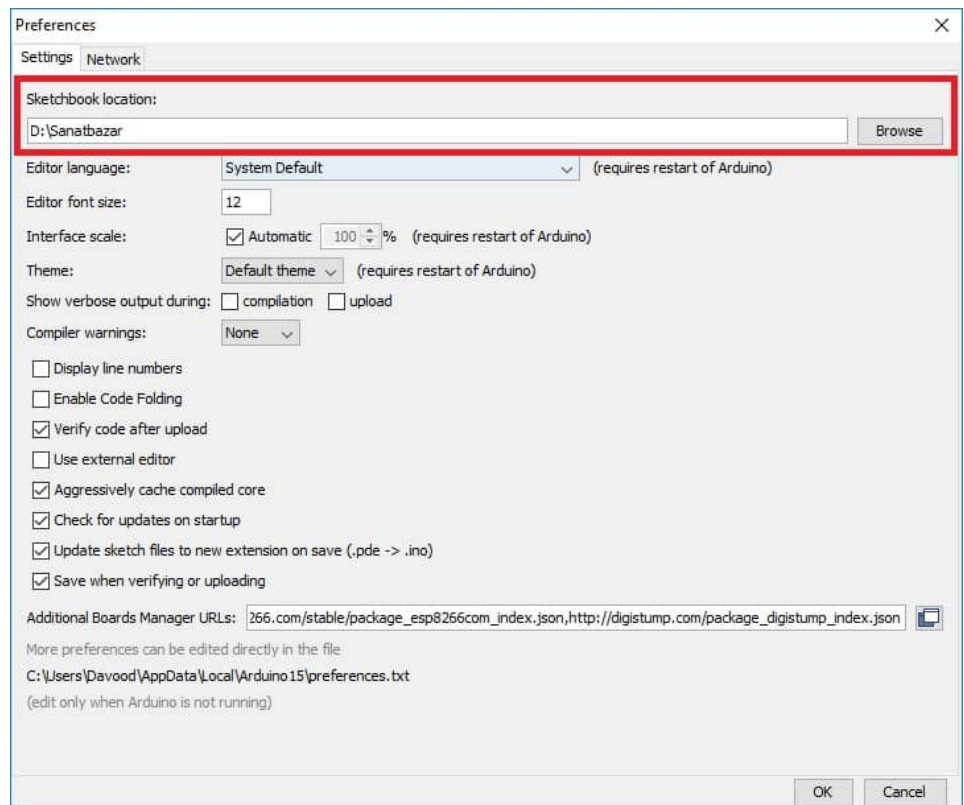
به دلیل متن‌باز بودن آردوینو، هر کسی می‌تواند برای آن، کتابخانه بنویسد. شما می‌توانید کتابخانه‌ای مخصوص کار خودتان بنویسید. به همین جهت ممکن است نیاز به کتابخانه‌ای پیدا کنید که در لیست نرم‌افزار موجود نیست. در این روش ابتدا فایل فشرده کتابخانه مورد نظرتان را دانلود کنید. این فایل می‌تواند کتابخانه‌های رسمی آردوینو باشد یا اینکه افرادی به صورت شخصی آن را نوشته و در اینترنت منتشر کرده‌اند. سپس در نرم‌افزار مسیر `Sketch > Include Library > Add .ZIP Library` را بروید. سپس فایل فشرده دانلود شده را انتخاب کنید. منتظر بمانید تا نرم‌افزار کتابخانه را نصب کند.



روش سوم

در این روش به صورت دستی اقدام به نصب کتابخانه می‌کنید. ابتدا فایل کتابخانه مورد نظر را دانلود کنید (یا شاید خودتان آن را نوشته‌اید!). پوشه شامل فایل‌های کتابخانه را در محل نصب کتابخانه‌های آردوینو کپی کنید. یک بار نرم‌افزار را ببندید و دوباره آنرا باز کنید. محل نصب کتابخانه‌های آردوینو را می‌توانید از مسیر `File > Preferences > Sketchbook location` پیدا کنید.





انجام اولین پروژه با آردوینو

تاکنون با سخت‌افزار و نرم‌افزار آردوینو آشنا شده‌اید. با چیزهایی که یاد گرفته‌اید می‌توانید یک پروژه ساده را انجام دهید. پیشنهاد من پروژه چراغ چشمک‌زن است. برای این پروژه من از آردوینو UNO استفاده کرده‌ام. در حین این پروژه ساده برخی دستورها و مفاهیم اولیه آردوینو را مطرح خواهیم کرد.

قطعات مورد نیاز:



عدد 1

[برد آردوینو و کابل رابط](#)



آردوینو را به رایانه وصل کنید و نرم افزار را باز کنید. نوع برد و پورت اتصال را از مسیر `Tools > Board` و `Tools > Port` انتخاب کنید. بر روی برد بیشتر آردوینوها، یک LED وجود دارد که به یکی از پین های دیجیتال متصل است. بنابراین می توانیم از آن به عنوان یک خروجی استفاده کنیم.



از آنجا که پین های دیجیتال قابلیت ورودی یا خروجی بودن را دارند، همیشه باید قبل از استفاده، نوع آن پین را تعیین کنید. این کار باید در `void setup` صورت گیرد. نحوه تعیین نوع پین با استفاده از دستور زیر انجام می شود:

```
pinMode(13, OUTPUT)
```

در داخل پرانتز، شماره پین و سپس نوع آن تعیین می‌شود. دقت کنید که کامپایلر آردوینو به حروف بزرگ و کوچک حساس است، پس دستورها را دقیقاً همان طور که مشخص شده‌اند بنویسید. نکته مهم دیگر این است که همیشه (به جز بعضی موارد خاص) باید در پایان هر خط، نقطه‌ویرگول ; بگذارید که به معنای پایان سطر است؛ در غیر این صورت با خطا روبرو خواهید شد. در صورتی که می‌خواستیم یک ورودی تعریف کنیم، دستور زیر را وارد می‌کردیم:

```
pinMode(13, INPUT)
```

یک برنامه‌نویس حرفه‌ای همیشه سعی می‌کند طوری کدنویسی کند که در صورت طولانی شدن برنامه، سردرگم نشود. مثلاً در این پروژه بهتر است یک متغیر به نام LED تعریف کنید و هر زمان که به پین LED نیاز داشتید، از این متغیر استفاده کنید. در برنامه‌نویسی آردوینو، همیشه قبل از استفاده از هر متغیری باید آنرا تعریف کنید. برای آشنایی با انواع متغیرها می‌توانید به [سایت آردوینو](#) مراجعه کنید. در اینجا من یک متغیر از جنس اعداد صحیح تعریف می‌کنم:

```
int LED=13
```

هر داده‌ای برای ذخیره در کامپیوتر باید یک قالب مشخص داشته باشد. به این قالب‌ها، نوع متغیر می‌گویند. نوع متغیر، برای کامپیوتر تعیین می‌کند که داده را به چه شکلی ذخیره کند. انواع مختلفی برای متغیرها وجود دارد، مثل: عدد صحیح (int)، کاراکتر (char)، عدد اعشاری (float)، عدد 8-بیتی (byte) و غیره. برای هر کدام از انواع متغیرها، نحوه ذخیره‌سازی و تعداد بیت اشغال‌شده توسط آن متغیر متفاوت است.

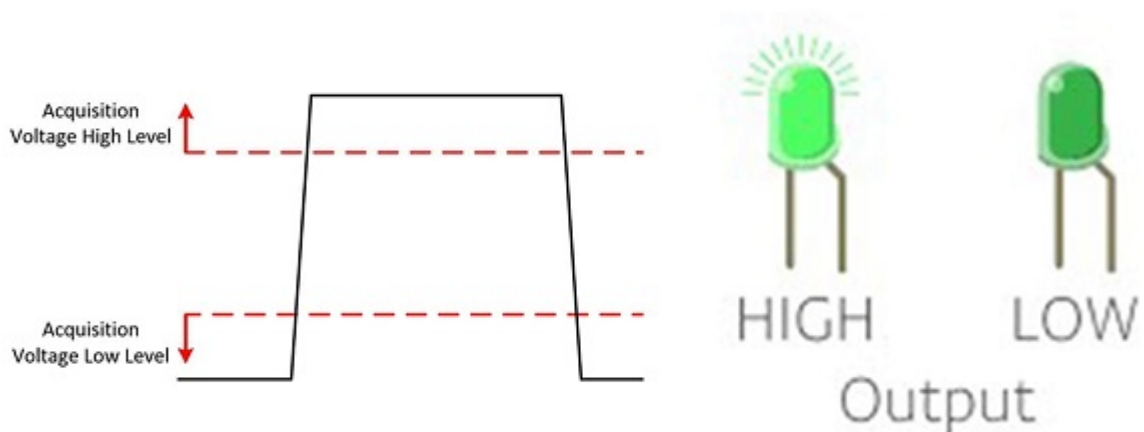
برای تعریف متغیر باید نوع آن را تعیین کنید، اما مقداردهی به آن اختیاری است. هرچند که در این مثال ما نیاز به مقداردهی داریم. معنی دستور بالا این است که نرم‌افزار یک متغیر به نام LED و از جنس عدد صحیح تعریف کرده و به محض برخورد با کلمه LED مقدار 13 را به جای آن قرار می‌دهد. این متغیر در طول برنامه قابل تغییر نیز هست.

می‌خواهیم یک LED داشته باشیم که مرتباً به مدت 1.0 ثانیه روشن و خاموش شود. برای این کار می‌خواهیم از پین‌های دیجیتال آردوینو استفاده کنیم. دقت کنید که متغیر دیجیتال تنها یکی از دو مقدار High یا 1 و Low یا 0 را می‌پذیرد که به معنی روشن یا خاموش است. وقتی که پین دیجیتال در حالت LOW است هیچ ولتاژی اعمال نمی‌کند در حالی که وقتی در وضعیت HIGH است ولتاژ حداکثر (در مورد آردوینو UNO این مقدار 5V است) را اعمال می‌کند. برای ارسال یک فرمان به پین دیجیتال از دستور زیر استفاده می‌کنیم:

```
digitalWrite(LED, HIGH)
```

یا

```
digitalWrite(LED, LOW)
```



همچنین اگر بخواهیم مقداری را از یک پایه دیجیتال بخوانیم از دستور زیر استفاده می‌کنیم:

```
digitalRead(LED)
```

دستور دیگری که در اینجا نیاز داریم، فرمان تاخیر است. وقتی که دستور تاخیر اجرا می‌شود، آردوینو به اندازه مدت زمانی که ما تعیین می‌کنیم صبر می‌کند و در این مدت تمام متغیرها را در وضعیت قبلی خود نگه می‌دارد. این دستور بر حسب میلی‌ثانیه محاسبه می‌شود. نحوه نوشتن دستور تاخیر به صورت زیر است:

```
delay(100)
```

حالا می‌توانید با زدن دکمه Upload برنامه را بر روی آردوینو بریزید. منتظر بمانید تا پیغام Done Uploading مطابق شکل زیر نشان داده شود.

```

Blink_LED | Arduino 1.8.9
File Edit Sketch Tools Help
Blink_LED
1 | /*
2 | SanatBazar
3 | Arduino Tutorial Series
4 | Author: Davood Dorostkar
5 | Website: www.sanatbazar.com
6 |
7 | */
8 |
9 | int LED=13;
10| void setup() {
11|   pinMode(LED, OUTPUT);
12| }
13|
14| void loop() {
15|   digitalWrite(LED, HIGH);
16|   delay(100);
17|   digitalWrite(LED, LOW);
18|   delay(100);
19| }

Done compiling.
Sketch uses 940 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for local variables.
Arduino/Genuine Uno on COM3

```

در صورتی که خطایی در برنامه رخ دهد، در قسمت پائین نرم‌افزار می‌توانید جزئیات آنرا مشاهده کنید. دلایل زیادی در به وجود آمدن خطا تاثیر دارد. معمول‌ترین آنها، اشتباه نوشتن دستورها یا عدم انتخاب درست تنظیمات برد و پورت است.

با استفاده از برنامه زیر می‌توانید پروژه چشم‌ک‌زن را اجرا کنید:

```

/*
SanatBazar
Arduino Tutorial Series
Author: Davood Dorostkar
Website: www.sanatbazar.com

*/

int LED=13;

void setup() {
  pinMode(LED, OUTPUT);
}

void loop() {
  digitalWrite(LED, HIGH);
  delay(100);
  digitalWrite(LED, LOW);
  delay(100);
}

```

نتیجه گیری

تبریک؛ به دنیای آردوینو خوش آمدید! شما موارد ساده مورد نیاز برای کار با آردوینو را یاد گرفتید. در این آموزش با نحوه نصب نرم افزار آردوینو و بخش های مختلف آن آشنا شدید و یاد گرفتید چطور یک پروژه را اجرا کنید. چیزهایی که می توانید در آردوینو یاد بگیرید بی پایان است، با این وجود تا اینجا شما اطلاعات اولیه را در اختیار دارید و آماده یادگرفتن هستید.

در آموزش بعدی، [نحوه راه اندازی و کار با سنسورهای مختلف دما و رطوبت](#) را خواهید آموخت.

نظرات شما باعث بهبود محتوای آموزشی ما می شود. اگر این آموزش را دوست داشتید، همین طور اگر سوالی در مورد آن دارید، از شنیدن نظراتتان خوشحال خواهیم شد.