

بسمه تعالی

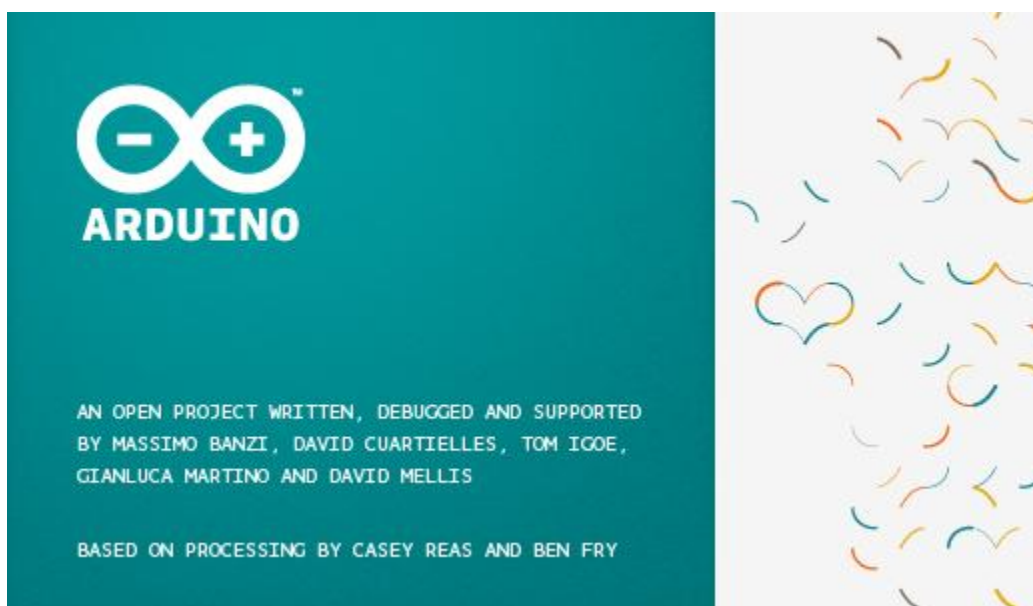


دانشگاه صنعتی شریف

دانشکده مهندسی برق - دانشکده مهندسی مکانیک

## آردوینو

معرفی و آموزش نحوه راه اندازی



آردوینو چیست ؟



آردوینو در حقیقت یک بستر سخت افزاری و نرم افزاری برای پیاده سازی بردهای الکترونیکی متنوع است که هدف اصلی آن آسان کردن ساخت بردها و کدنویسی آنها بوده و به صورت کاملاً **open-source** در اختیار عموم قرار گرفته است. آردوینو با آماده ساختن بسیاری از ماژولهای کاربردی و نیز ساخت یک بستر کدنویسی ساده و روان، یک طراح الکترونیک را از پیچیدگیهای اضافی مثل راه اندازیهای اولیه و سیمکشیهای زیاد و یا نوشتن تمام توابع پایه‌ی مورد احتیاج برنامه-نویسی سخت افزار بی‌نیاز می‌کند. انواع ماژولها از ماژول-های ساده فقط برای راه اندازی میکروکنترلر گرفته تا شیلدهای درایو موتور و حتی اتصال به شبکه‌ی وای فای در آردوینو بصورت از پیش طراحی شده موجود است که می‌تواند به پروژه‌های شما اضافه شود. همچنین کد نویسی برای آردوینو به زبان آردوینو است که بسیار مشابه زبان C++ می‌باشد و در محیط کامپایلر آردوینو قابل کامپایل و پروگرام کردن روی برد است و نکته‌ی بسیار قابل توجه در مورد آن این است که بسیاری از توابع پایه‌ای مورد نیاز در کد زدن شامل توابع ریاضی مثل سینوس و ... و یا توابع کار با زمان و نیز توابع کار با امکانات جانبی میکرو مثل مبدل آنالوگ به دیجیتال یا پورت سریال و... در آن با سادگی بسیار زیاد حتی بسیار ساده تر از محیطی مثل کدویژن پیاده سازی شده است که این امر راحتی و نیز حجم کدنویسی را به شدت کاهش می‌دهد.

در ادامه ابتدا به معرفی تنها یکی از بردهای نسبتاً پرکاربرد آردوینو که در دو آزمایش مطرح شده در درس الکترونیک کاربردی استفاده خواهد شد می‌پردازیم، سپس محیط



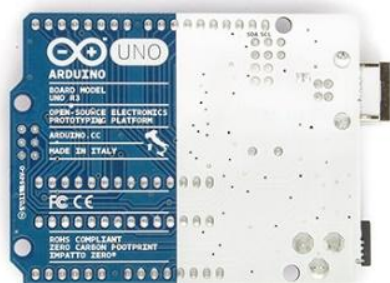
و مقدمات کدنویسی در بستر آردوینو را مورد بررسی قرار می‌دهیم .

سایت شرکت آردوینو :

<http://arduino.ir>

<http://arduino.cc>

برد Arduino Uno :



برد آردوینو Uno یک میکروکنترلر بر پایه ATmega328 می‌باشد (دیتاشیت ATmega328 را دانلود کنید و نگاهی اجمالی بر آن بیندازید). در حقیقت این برد چیزی جز یک میکروکنترلر ATmega328 نیست که تنها در آن کار راه‌اندازی میکرو ساده شده و مدارات جانبی مورد نیاز برای راه‌اندازی مثل نوسان ساز ، مدار ریست و ... پیاده‌سازی شده است و استفاده کننده هیچ درگیری برای عملیات زمان‌بر برای راه‌اندازی میکرو ندارد. پین‌های مختلف میکرو در قالب دو هدر در دو طرف برد به راحتی در دسترس قرار گرفته است. این برد ۱۴ پین ورودی و خروجی دیجیتال (که ۶ تای آن می‌تواند به عنوان خروجی PWM استفاده گردد) ، ۶ ورودی آنالوگ، یک نوسان-ساز سرامیکی ۱۶ مگاهرتز (به عنوان کلاک) ، یک پورت USB ، یک پاورجک (ورودی منبع تغذیه) ، یک ICSP header و یک دکمه ریست دارد. برد Uno شامل کلیه امکانات مورد نیاز جهت بکارگیری



میکروکنترلر موجود بر روی برد می باشد. برای شروع تنها با یک کابل USB، به سادگی برد را به کامپیوترتان متصل کنید و یا آن را با یک آداپتور AC-To-DC و یا باتری راه اندازی نمایید. ما در این دو آزمایش فقط با پورت USB کار خواهیم کرد. این اتصال هم به عنوان منبع تغذیه برد کار می کند و هم پروگرام کردن را به راحتی انجام می دهد.

### خلاصه

میکروکنترلر	۳۲۸ATmega
ولتاژ عملیاتی	۵ ولت
ولتاژ (پیشنهادی) ورودی	۷-۱۲ ولت
ولتاژ ورودی (محدوده)	۶-۲۰ ولت
پین های دیجیتال ورودی/خروجی	۱۴ (۶ تای آن به عنوان خروجی PWM استفاده می شود.)
پین های ورودی آنالوگ	۶
جریان DC هر پین ورودی و خروجی	۴۰ میلی آمپر
جریان DC جهت پین ۷۳,۳- پین تغذیه	۵۰ میلی آمپر
حافظه فلش	۳۲ کیلوبایت (۳۲۸ATmega) که ۰,۵ کیلوبایت از آن مورد استفاده BootLoader قرار می گیرد.
SRAM	۲ کیلوبایت (۳۲۸ATmega)
EEPROM	۱ کیلوبایت (۳۲۸ATmega)
سرعت ساعت	۱۶ مگاهرتز



## پین‌های ورودی و خروجی :

هریک از ۱۴ پین دیجیتال Uno می تواند با استفاده از توابع [pinMode\(\)](#)، [digitalWrite\(\)](#)، [digitalRead\(\)](#) به عنوان ورودی یا خروجی استفاده شود. ولتاژ پین ها ۵ ولت بوده و ظرفیت جریان جهت هر پین حداکثر ۴۰ میلی آمپر می باشد. همچنین هر یک از این پین ها دارای یک مقاومت داخلی (۲۰-۵۰ کیلو اهم) جهت Pull-Up می باشد (که به صورت پیش فرض غیرفعال است).

بعلاوه بعضی از پین ها دارای عملکردهای منحصر به فردی می باشند که شرح آن در ذیل آمده است:

- Serial - 0 (RX) و 1 (TX) : پین RX برای دریافت و TX جهت انتقال اطلاعات به صورت سریال و با پروتکل TTL استفاده می شود.

- External interrupts (وقفه های خارجی) - ۲ و ۳: این پین ها می توانند طوری تنظیم شوند که یک وقفه را براساس اندکی افزایش یا کاهش لبه، و یا هر نوع تغییر در مقدار، ایجاد نمایند. برای جزئیات بیشتر، تابع [attachInterrupt\(\)](#) را مشاهده نمایید.

- PWM - ۳, ۵, ۶, ۹, ۱۰, ۱۱ : امکان دسترسی به یک خروجی PWM هشت بیتی را با استفاده از تابع [analogWrite\(\)](#) فراهم می کنند.

- SPI - 10(SS), 11(MOSI), 12(MISO), 13(SCK): با استفاده از توابع کتابخانه ای SPI این پین ها می توانند یک ارتباط [SPI library](#) ایجاد نمایند. (SPI یک پروتوکل ارتباط



سریال برای اتصال میکرو به میکرو یا دیگر یا سایر وسایل  
(جانبی است)

• **LED-13**: یک LED آماده، به پین دیجیتال ۱۳ متصل شده است. هنگامی که پین در حالت HIGH قرار دارد، LED روشن و زمانی که پین در حالت LOW قرار دارد، خاموش می شود.

Uno ، ۶ ورودی آنالوگ دارد که از ۰A تا ۵A نامگذاری شده اند. میزان تفکیک پذیری هر یک از پین ها تا ۱۰ بیت می باشد (به عنوان ۱۰۲۴ مقدار مختلف یعنی بازه ۰ تا ۵ ولت را به ۱۰۲۴ قسمت تقسیم می کنند). به صورت پیش فرض این پین ها می توانند ولتاژی بین ولتاژ پایه (Ground) تا حداکثر ۵ ولت را اندازه گیری نمایند. ولی با استفاده از پین AREF و تابع [analogReference\(\)](#) تغییر حد بالای میزان تفکیک پذیری امکان پذیر می باشد. همچنین بعضی از پین ها دارای عملکردهای منحصر به فردی می باشند که شرح آن در ذیل آمده است:

• **TWI**: پین ۴A یا SDA و ۵A یا SCL: این پین ها امکان ایجاد یک ارتباط TWI را با استفاده از [توابع کتابخانه ای Wire](#) مقدور می سازند.

سایر پین ها:

• **AREF**: ولتاژ مرجع برای ورودی های آنالوگ، از طریق این پین و با استفاده از تابع [analogReference\(\)](#) تأمین می گردد.

• **Reset**: وضعیت لاین مرتبط را برای ریست میکروکنترلر در حالت Low قرار می دهد، معمولاً زمانی از این پین استفاده می شود که بخواهید بر روی شیلدتان دکمه ریست



قرار دهید. زیرا استفاده از شیلدها از دسترسی به دکمه ریست موجود بر روی برد آردوینو جلوگیری می کند.

برای شروع کار با Uno کافی است آن را با کابل USB به کامپیوتر خود متصل نموده و بعد از اتصال یک بار دکمه قرمز رنگ موجود روی برد را فشار دهید تا برد ریست شود.

## برنامه نویسی آردوینو :

برای شروع کار برنامه نویسی در محیط آردوینو برای بردهای این شرکت ابتدا باید کامپایلر آن را دانلود کنید . برای این کار از لینک زیر استفاده کنید و یا از سرور دانشکده مکانیک آن را بردارید.

### برای ویندوز      برای مک

توجه داشته باشید که درایورهای مربوطه برای شناخت بردهای آردوینو همراه فایل نصبی فوق ، نصب می شود و لذا بعد از نصب برنامه اگر یک برد آردوینو مثلا Uno را به کامپیوتر خود وصل کنید باید کامپیوتر آن را بشناسد. اگر شناخت مراحل زیر را دنبال کنید :

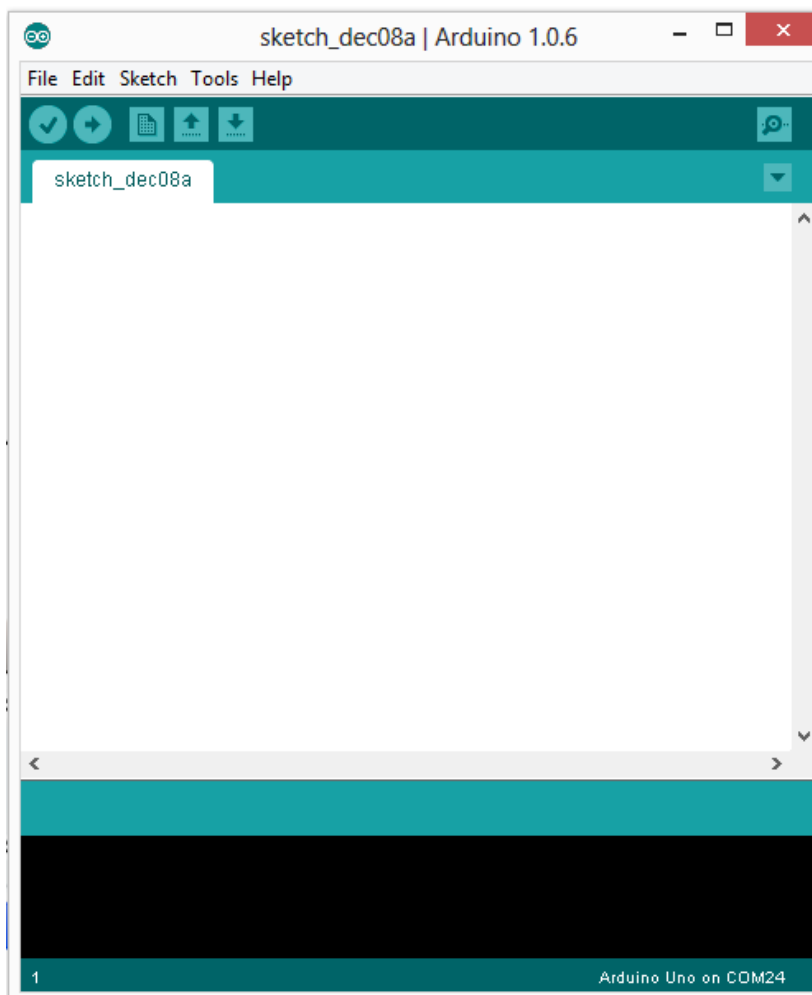
- روی منوی Start کلیک کرده و Control Panel را باز نمایید.
- در پنجره Control Panel به System and Security بروید. سپس، روی System کلیک کنید. زمانی که پنجره System بالا آمد، Device Manager را باز کنید.



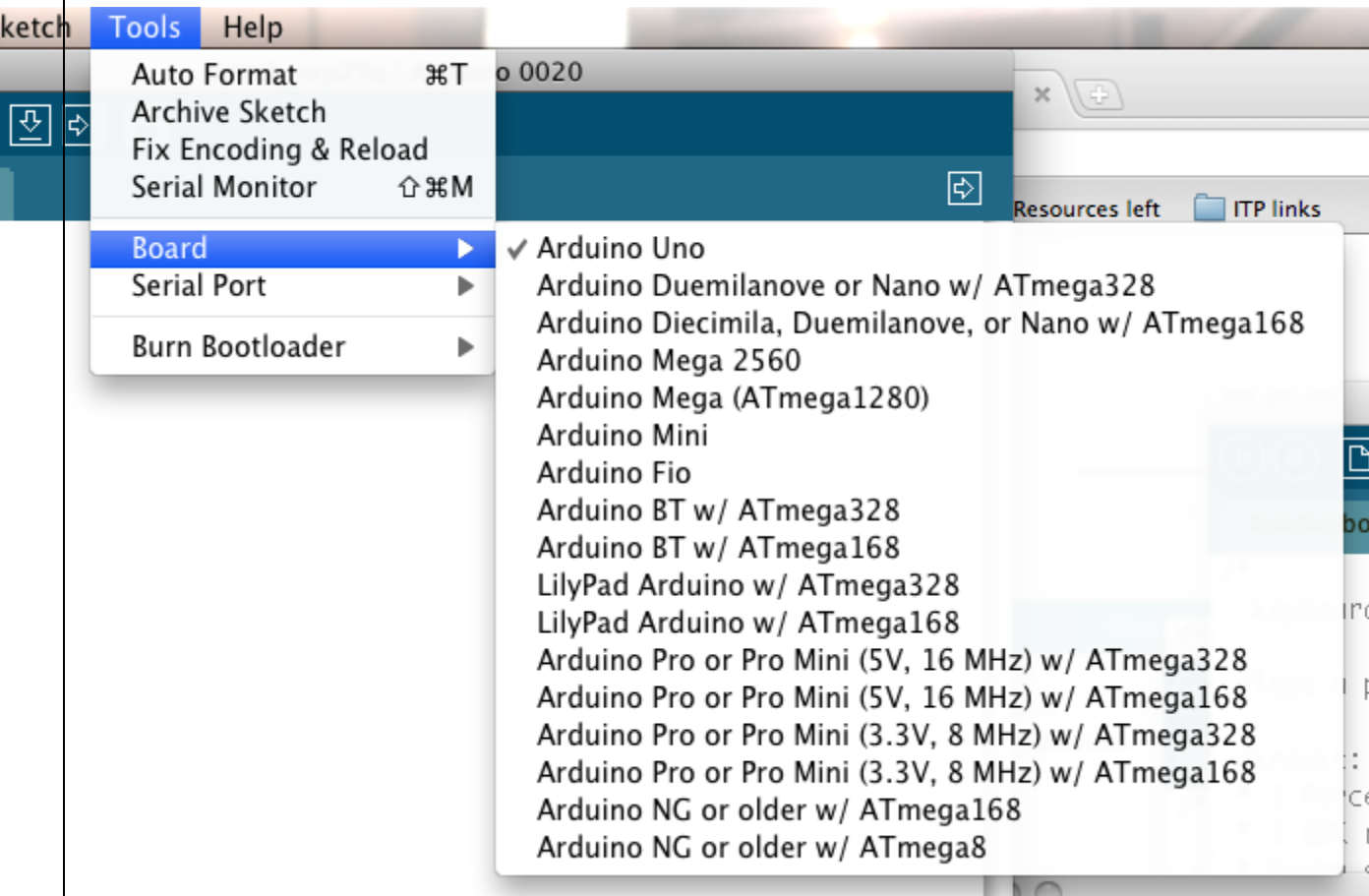
- زیر Ports (Com & LPT) را نگاه کنید. باید یک پورت باز به نام "Arduino UNO (COMxx)" ببینید.
- روی پورت "Arduino UNO (COMxx)" راست کلیک کنید و گزینه "Update Driver Software" را انتخاب نمایید.
- سپس، گزینه "Browse my computer for Driver software" را انتخاب کنید.
- در نهایت، به مسیر برنامه رفته و فایل راه انداز را به نام "arduino.inf" که در پوشه "Drivers" نرم افزار آردوینو قرار دارد، انتخاب نمایید. ( زیرشاخه "FTDI USB Drivers") . اگر از یک نسخه قدیمی محیط برنامه نویسی ( ۱,۰,۳ یا قدیمی تر) استفاده می کنید، فایل راه انداز Uno را با نام "Arduino UNO.inf" انتخاب کنید.

پس از نصب برنامه و نیز اتصال برد به کامپیوتر (چراغ سبز رنگ مربوط به منبع که با برچسب PWR مشخص شده است باید روشن شود) با محیط زیر مواجه می شوید :





در مرحله‌ی بعدی باید از بین برده‌های آردوینو مختلف موجود ، برد خود را که به کامپیوتر متصل کرده و قصد کار با آن را دارید مشخص کنید آن را از منوی **Tools > Board** انتخاب کنید.(شما همان گزینه اول یعنی **Arduino Uno** انتخاب نمایید)

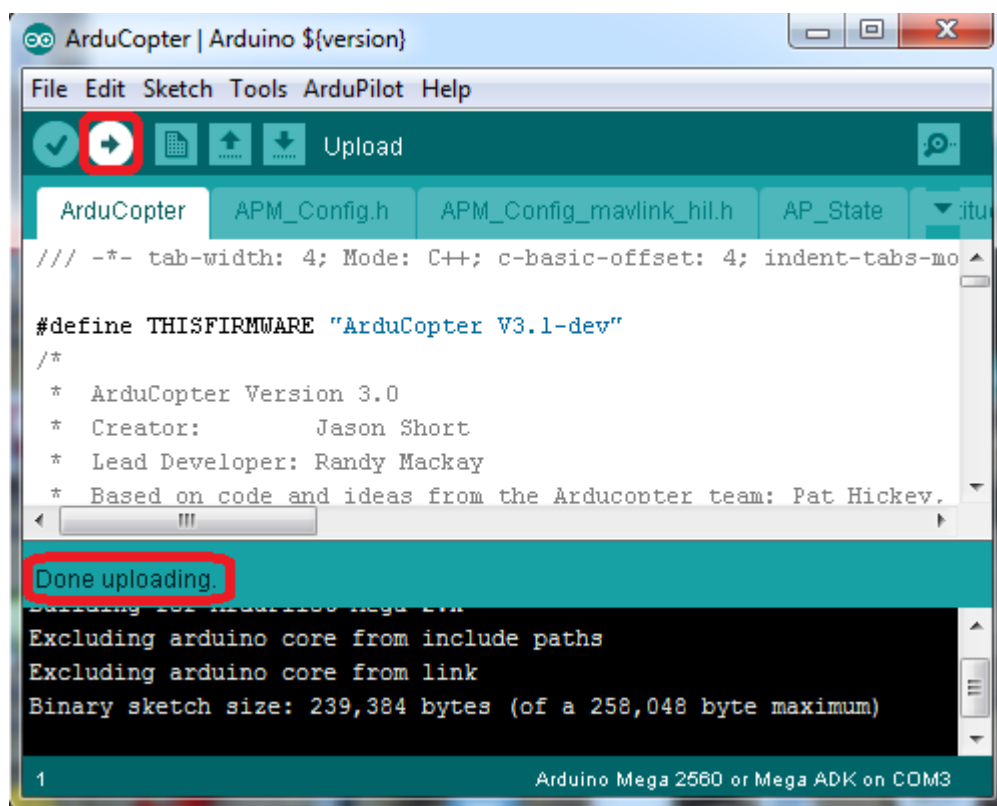


سپس باید پورت سریالی که کامپیوتر تحت آن آردوینو را شناخته معین کنید. پورت سریال برد آردوینو خود را از منوی **Tools > Serial Port** انتخاب کنید. این گزینه احتمالاً **COM3** یا بالاتر است (**COM1** و **COM2** معمولاً برای پورت های سریال سخت افزاری رزرو شده اند). برای دانستن پورت مربوط به آن، شما می توانید برد آردوینو خود را از کامپیوتر بیرون کشیده و مجدداً منو را باز نمایید؛ پورتی که این بار نشان داده نمی شود، برد آردوینو است. برد را مجدداً متصل کنید و آن پورت سریال را انتخاب نمایید. (یا از **Device manager** کامپیوترتان آن را پیدا کنید)



در مرحله آخر از منوی **Tools > Programmer** نوع پروگرامر را **Arduino as ISP** انتخاب کنید.

حال می‌توانید هر کدی که نوشته اید را با فشردن دکمه‌ی **upload** روی برد خود پروگرام نمایید. توجه داشته باشید که اگر عمل پروگرام کردن به درستی انجام شود مطابق تصویر زیر با پیغام **Done Uploading** مواجه می‌شوید وگرنه در قسمت مشکی رنگ پایین برنامه با پیغام خطا مواجه خواهید شد.



## نحوه‌ی کدنویسی در آردوینو :

به طور کلی می‌توان گفت ادبیات کدنویسی در آردوینو تا حد بسیار زیادی مشابه زبان C++ می‌باشد. نکته‌ی قابل اتکا در



مورد آردوینو این است که یک راهنمای بسیار جامع و کامل از مبتدی تا تمام کلاسها و توابع حرفه‌ای ارائه داده که برای هر کسی با هر سطح اطلاعات کاملاً کارگشااست. این راهنما از طریق سایت آردوینو بخش **Learning > Reference** و یا داخل برنامه‌ی آردوینو منوی **Help > Reference** قابل دسترسی و مطالعه است. همچنین در منوی **File > Example** در نرم افزار یا بخش **Learning > Example** در سایت تعداد زیادی مثال نوشته شده که بسیار نیز مفید و کاربردی هستند موجود است، برای استفاده از تمام قسمت‌های میکرو کدهای متعدد موجود است که برای آشنایی می‌توانید روی آن‌ها کار کنید.

ما در اینجا به عنوان نمونه به چند تابع مهم‌تر که در آزمایش‌ها نیز مورد نیاز است اشاره می‌کنیم :

کدهای آردوینو شامل دو بخش اصلی است یکی تابع **void setup()** که در آن مقاداردهی‌های اولیه مثل راه‌اندازی درگاه سریال و مقاداردهی‌های اولیه به متغیرها و ... انجام می‌شود این تابع تنها یکبار هنگام شروع کار میکرو (روشن شدن یا ریست شدن) اجرا می‌شود.

تابع اصلی دیگر تابع **void loop()** است که این تابع مشابه **while(1)** در **code vision** است، این تابع در طی کار میکرو مداماً صدا شده و کدهای درون آن اجرا می‌شود که اصل کار میکرو است.

تابع **pinMode(pin num , output or input)** نوع پین‌ها را می‌تواند تعیین کند.

تابع **analogRead (pin num)** از ورودی‌های آنالوگ مقدار آنالوگ را خوانده و تبدیل به دیجیتال (عددی بین ۰ تا ۱۰۲۴) می‌کند.



تابع `digitalWrite(pin num , HIGH or LOW)` یک پین را صفر یا یک می-کند.

تابع `delay(num)` به اندازه `num` با واحد میلی ثانیه تاخیر ایجاد می‌کند.

تابع `analogWrite(pin num , value)` یک موج PWM که `duty cycle` آن را `num` مشخص می‌کند مروی پین هایی که خاصیت PWM را دارند ایجاد می‌کند .

\* توجه داشته باشید که روی برد `Uno` فقط آن پین‌هایی که در کنار شماره‌شان علامت `~` دارند قابلیت تولید PWM را دارا هستند.

\* لازم به ذکر است هیچ یک از توابع فوق نیازی به `include` کردن کتابخانه‌ی خاصی ندارند.

\* برای کسب اطلاعات کاملتر و دقیقتر در مورد توابع فوق و دیگر توابع مورد استفاده در کد نویسی به `reference` آردوینو مراجعه کنید.