# آشنایی با فضای برنامه نویسی

برای اینکه بتوانیم کد های مورد نیازمان را بنویسیم، اول باید با فضای برنامه نویسی آردینو آشنا بشویم. در بخش معرفی آردینو با این فضا و نحوه‌ی چک کردن یک برد آشنا شدیم. حال به بررسی دقیق تر این فضا می‌پردازیم. شما در این فضا می‌توانید منطقی را که از میکروکنترلر خود انتظار دارید به وسیله کد C++ به میکروکنترلر خود بدهید. ما ابتدا کمی با ابزار های این فضا آشنا می‌شویم و سپس اولین کدمان را می‌نویسیم.

# منو فضای برنامه نویسی

این منو شامل 2 بخش است. منوی اول شامل بخش هایfile ،edit ،sketch ، tools و help می‌باشد.



## file

قسمت file مانند تمامی برنامه ها مکانی برای انجام عملیات روی فایل می‌باشد. در بخش معرفی آردینو شما با یکی از زیر بخش های فایل به نام مثال ها (examples) آشنا شدید. در این بخش مثال هایی از کد های کاربردی و یا دستورات اصلی و ابتدایی آردینو به صورت پیش فرض توسط شرکت آردینو آورده شده است.

## edit

در قسمت edit ابزار هایی که برای ایجاد تغییرات در کد مورد نیاز هستند مانند copy، cut و غیره. را می‌توانید پیدا کنید.

## sketch

در قسمت sketch شما می‌توانید به دستور هایی که روی کد صورت می‌پزیرد دسترسی پیدا کنید. مانند عملیات آپلود و عملیات verify که شما باید قبل از اینکه کد را به داخل برد آپلود کنید، از درست بودن نوشتار کد مطمئن شوید. این کار توسط دستور verify صورت می‌پذیرد. البته لازم به ذکر است که به هنگام زدن دکمه آپلود، نرم افزار به صورت خودکار کد دستور verify را در پشت صحنه اجرا می‌کند و در صورت نبود مشکلات نوشتاری، کد شما را آپلود می‌کند. یکی دیگر از زیربخش ها included library یا همان کتابخانه است. شما در این بخش می‌توانید کتابخانه هایی را که در دستگاه خود ذخیره دارید را ببینید. کتابخانه ها ابزار هایی بسیار مناسب برای راحت سازی برنامه نویسی هستند. در بخش های آینده از کتابخانه های مختلف استفاده خواهیم کرد.

## help

در بخش help شما می‌توانید از منابعی که آردینو برای شما فراهم کرده است استفاده کنید. یکی از این منابع reference می‌باشد که در آن تمامی دستور ها و تعاریف مربوط به فضای برنامه نویسی آردینو آمده است. این منبع به صورت پیش فرض در برنامه وجود دارد و شما می‌توانید بدون وصل بودن به اینترنت از آن استفاده کنید.

## tools

بخش tools بخش بسیار مهمی است. ما به بررسی قسمت های سریال مانیتور (serial monitor)، سریال پلاتر (serial plotter)، برد (board) و پورت (port) از این بخش می‌پردازیم. بقیه موارد نیز به مرور زمان گفته خواهند شد.

## سریال مانیتور

در بعضی از موارد شما می‌خواهید از طریق یک پورت سریالی مانند usb کامپیوتر، داده هایی را به آردینو بفرستید و یا اینکه داده هایی را از آردینو بگیرید. مثلا می‌خواهید بدانید که اگر درب خانه باز شد به دستگاه شما پیامی ارسال شود که در آن به باز شدن درب خانه اشاره شده باشد. این ارتباط بین دستگاه و آردینو که از طریق سیم می‌باشد تحت عنوان serial monitor در آردینو معرفی شده است. در 3 بخش مجزا ما به بررسی این ابزار قدرتمند خواهیم پرداخت.

## سریال پلاتر

در بعضی از موارد شما نیاز دارید داده هایی را که توسط سنسور ها می‌خوانید، تحلیل کنید. برای این کار معمولا داده ها به صورت یک نمودار نمایش داده می‌شوند تا تحلیل آن ها ساده تر باشد. به طور مثال شما می‌خواهید دمای یک اتاق را نسبت به زمان رسم می‌کنید. آردینو برای این مسئله ابزاری را به نام serial plotter معرفی کرده که کار را بسیار ساده می‌کند. شما می‌توانید به وسیله این دستور داده هایی را که از آردینو به رایانه خود می‌فرستید را در واحد زمان به صورت یک نمودار نمایش دهید.

## برد

شرکت آردینو برد هایی بسیار زیادی دارد. در بخش معرفی آردینو تعدادی از برد های اصلی آردینو و نحوه انتخاب آنها بیان شد. حال با توجه به گفته های قبلی ما، میکرو کنترلر های این برد ها با یک دیگر فرق می‌کنند برای همین به هنگام آپلود کردن کد به آنها، منطقا نباید از یک الگوریتم استفاده کنیم. برای همین قبل از آپلود کردن کد، شما باید به وسیله بخش tools/board برد مورد استفاده خود را انتخاب کنید. اگر این کار را انجام ندهید و برد اشتباه انتخاب شده باشد، شما خطای "Error compiling for board" را دریافت خواهید کرد.



یا در بعضی از مواقع خطای "An error occurred while uploading the sketch" خواهید گرفت. این یک خطای مشترک است. در حالت کلی وقتی این خطا ظاهر می‌شود که مشکلی در آن طرف سیم (برد) وجود داشته باشد.



در نظر داشته باشید که اگر به هنگام آپلود کردن مشکلی برای سیم پیش بیاید و از وصل شدن ارتباط بین برد و رایانه جلوگیری کنید شما خطای "avrdude: ser\_send(): write error: sorry no info avail" دریافت خواهید کرد.



## پورت

برد آردینو از طریق پورت usb به دستگاه شما متصل شده است. حال فرض بفرمایید به هنگام آپلود کردن کد به جای یک برد آردینو 2 برد آردینو مختلف به دستگاه شما وصل باشد. حال نرم افزار چگونه متوجه شود که شما می‌خواهید این کد بخصوص را به کدام یک از این برد ها بفرستید؟ برای اینکه این مشکل پیش نیاید شما باید آدرس usb را که برد مورد نظر به آن وصل شده است را برای نرم افزار مشخص کنید. به صورت پیش فرض ویندوز آدرس هر یک از usb ها را تحت هنوان port ذخیره می‌کند. مثلا ممکن است آردینو uno به Port1 وصل باشد و آردینو mega به Port7 وصل باشد. شما می‌بایست از طریق tools/port پورت مورد نظر خود را انتخاب کنید. اگر این کار را نکنید شما خطای خطای "An error occurred while uploading the sketch" را دریافت خواهید کرد.



## منوی ثانویه

حال به بررسی منوی ثانویه یا دسترسی سریع می‌پردازیم. در این منو شما می‌توانید به گزیده‌ای از ابزار ها که در قسمت قبل به آنها اشاره کردیم، به صورت سریع دسترسی پیدا کنید. اگر نشانگر موس خود را بر روی هر یک از شکل های این نوار ببرید، نام ابزار آن در سمت راست نوشته خواهد شد.

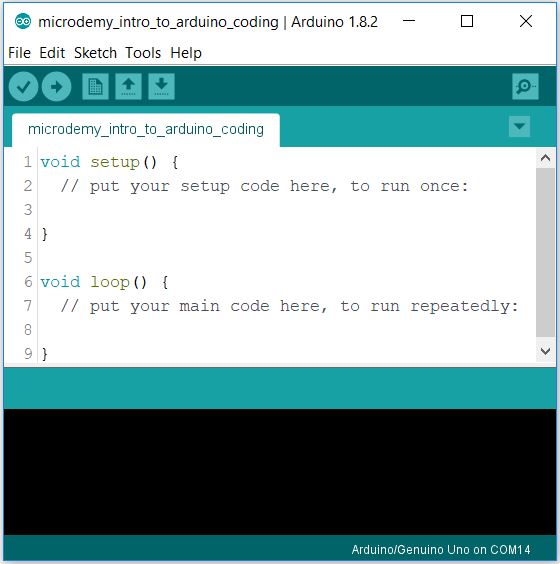


بسیار مناسب است که برای بالا رفتن سرعت یا از منوی دسترسی سریع استفاده کنید یا از shortcut های موجود در نرم افزار بهره ببرید. تمامی shortcut ها به صورت زیر هستند:



# محل کد نویسی

در این بخش شما کد مورد نظرتان را در دو بخش ستاپ (setup) و حلقه (loop) می‌نویسید. در واقع کد مجموعه‌ای از دستورات است که به صورت خط به خط اجرا می‌شود و ابتدا و انتهایی دارد. البته این تعریفی ساده برای کد های ساده می‌باشد. کد هایی که به منظور کنترل یک دستگاه زده می‌شوند از منطق بخصوصی برخوردار هستند. در پروژه های کنترلی هدف همواره کنترل یک پارامتر می‌باشد. به طور مثال شما می‌خواهید توسط یک فن، دمای هوای داخل یک محفظه را ثابت نگه دارید. برای این کار شما باید دما را توسط یک سنسور دما به صورت پیوسته بخوانید تا اگر دما از دمای مد نظر بالاتر رفت فن روشن شود و وقتی پایین تر آمد فن خاموش شود. پس کنترلر شما باید به صورت پیوسته دما را بخواند و دستورات متناسب با آن شرایط را بدهد. به بیان دیگر باید در یک حلقه بسته تا بینهایت دما را بخواند و دستور بدهد.



## void loop ( )

در این بخش دستوراتی که باید به صورت پیوسته و مداوم توسط کنترلر اجرا شود مانند خواندن دما، نوشته می‌شود. همان طور که از اسم تابع مشخص است یک حلقه می‌باشد که تا بینهایت اجرا می‌شود.

## void setup ( )

این بخش از کد فقط یک بار اجرا می‌شود و آن هم در ابتدای روشن شدن میکروکنترلر می‌باشد. در این بخش دستور هایی مانند نام پین ها، وجود و یا عدم وجود نمایشگر، اطلاعات مربوط به صفحه کلید و غیره را که باید فقط یک بار اجرا شوند، را می‌نویسیم. همان طور که از اسم این تابع مشخص است وظیفه این تابع آماده کردن کنترلر برای شروع عملیات کنترلی می‌باشد.

# چشمک زن

هدف از این بخش نوشتن کدی است که دیود نوری داخل برد آردینو را به صورت پیوسته روشن و خاموش کند. در بخش معرفی آردینو ما به وسیله‌ی یکی از مثال های آردینو به نام چشمک زن این کار را انجام دادیم تا بتوانیم سالم بودن برد را بررسی کنیم. حال در اینجام می‌خواهیم خودمان این کد را بنویسیم. برای این کار ابتدا باید با دستورات زیر آشنا بشویم.

## پین های دیجیتال

همان طور که در بخش معرفی آردینو گفته شد، هر یک از برد های آردینو دارای پین های دیجیتالی هستند. به طور مثال برد Uno دارای 14 پین دیجیتال می‌باشد که هر کدام توسط یک عدد از 1 تا 14 نام گزاری شده‌اند. این پین ها توانایی خواندن مقدار دیجیتالی یک سیگنال و یا تولید یک سیگنال دیجیتال را دارند.

## pinMode ( )

این تابع حالت یک پین دیجیتال را مشخص می‌کند. یک پین دیجیتال یا یک سیگنال دیجیتالی تولید می‌کند و یا اینکه مقدار دیجیتالی یک سیگنال را می‌خواند. به بیان دیگر یا یک سیگنال دیجیتالی از آن خارج می‌شود که در این صورت حالت این پین خروجی (OUTPUT) است، و یا اینکه یک سیگنال دیجیتالی به آن وارد می‌شود که در این صورت حالت این پین ورودی (INPUT) می‌باشد.



توجه: حرف اول کلمه Mode به صورت بزرگ نوشته شده است و همین طور حالت پین به صورت تمام بزرگ نوشته می‌شود. نرم افزار C++ به بزرگ و کوچکی حروف حساس می‌باشد.

## digitalWrite ( )

این تابع یک وضعیت دیجیتالی یک پین را مشخص می‌کند. این تایع توسط یک متغیر باینری مقادیر دیجیتالی را بر روی پین می‌نویسد. اگر مقدار باینری 1 یا HIGH را بر روی پین بنویسید، ولتاژ پین 5 ولت می‌شود و اگر مقدار باینری 0 یا LOW را بر روی پین بنویسید، ولتاژ آن 0 ولت می‌شود. مقادیر HIGH و LOW مقادیر تعریف شده‌ای هستند که فقط در آردینو کاربرد دارند و نشانگر همان 0 و 1 باینری و دیجیتالی هستند.



توجه: باز هم حرف اول کلمه Write بزرگ نوشته شده است. همین طور وضعیت پین نیز به طور کامل بزرگ نوشته می‌شود. همیشه به یاد داشته باشید توابعی که 2 کلمه‌ای هستند، حرف اول کلمه دوم بزرگ نوشته می‌شود.

## delay ( )

این تابع برای مدت زمانی بر واحد میلی ثانیه، برنامه را متوقف می‌کند. ورودی این تابع مقدار توقف بر واحد میلی ثانیه می‌باشد.



توجه: این تابع به صورت کامل برنامه را در این خط نگه می‌دارید. این به این معنی است که در این هنگام میکروکنترلر هیچ دستور دیگری را اجرا نخواهد کرد که این مسئله بسیار مهم می‌باشد.

## پین دیود نوری داخلی

هر برد آردینو دارای یک دیود نوری بر روی برد خود می‌باشد. این دیود نوری به وسیله‌ی یک پین دیجیتال قابل دسترسی می‌باشد. همان طور که در بخش معرفی آردینو گفته شد، مشخصات هر برد در سایت آردینو آمده است ودر آن شمار پینی که این دیود نوری به آن وصل شده است نیز ذکر شده. لازم به ذکر است که در اکثر برد های آردینو این دیود به پین شماره 13 وصل شده است. شما می‌توانید با تعیین وضعیت این پین توسط تابع digitalWrite، وضعیت روشن یا خاموش بودن این دیود نوری را مشخص کنید. واضح است که دیود در وضعیت HIGH روشن خواهد بود و در وضعیت LOW خاموش خواهد بود.

## نوشتن کد مربوط به چشمک زن

فضای کد زنی دارای یک حلقه می‌باشد که به صورت مداوم تکرار می‌شود. حال اگر ما در این حلقه به ترتیب به دیود نوری دستور روشن، مکث، خاموش و مکث را بدهیم، در صورت تکرار این دستورات دیود نوری باید چشمک بزند. برای این کار به ترتیب دیود را با تابع digitalWrite روشن می‌کنیم سپس با تابع delay آن را روشن نگه می‌داریم و همین کار را برای خاموش کردن نیز انجام می‌دهیم.



حال می‌توانید با کم کردن مکث بین روشن و خاموش فرکانس روشن-خاموش شدن دیود را تغییر دهید. آن را به قدری کم کنید تا دیگر خاموش شدن دیود را نبینید و به سرعت عکس برداری چشمتان پی ببرید.

توجه: متغیر led که در ابتدای برنامه تعریف شد، مقدار عددی 13 را در خود نگه می‌دارد. نکته قابل توجه این می‌باشد که این متغیر از جنس متغیر های سراسری بوده که در تمامی کد قابل استفاده است. برای اطلاع بیشتر از این موضوع می‌توانید به بخش متغیر ها مراجعه کنید.