# آشنایی با فضای برنامه نویسی

برای اینکه بتوانیم کدهای 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 ابزارهای 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 است.



## File

قسمت File مانند تمامی برنامه‌ها مکانی برای انجام عملیات روی فایل است. در بخش معرفی آردینو شما با یکی از زیر بخش‌های فایل به نام مثال‌ها (Examples) آشنا شدید. در این بخش مثال‌هایی از کد‌های کاربردی و یا دستورات اصلی و ابتدایی آردینو به صورت پیش فرض توسط شرکت آردینو آورده شده است. البته گفتنی است که بعضی از این مثال‌ها ممکن است مربوط به کتابخانه‌هایی باشد که بر روی دستگاه شما نصب شده‌اند.

## Edit

در قسمت Edit ابزارهایی که برای ایجاد تغییرات در کد نیاز هستند مانند Copy، Cut و ... را می‌توانید پیدا کنید.

## Sketch

در قسمت Sketch می‌توانید به دستور‌هایی که روی کد صورت می‌پذیرد دسترسی پیدا کنید. مانند عملیات آپلود و عملیات Verify که 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 گفتنی 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 ببینید. کتابخانه‌ها ابزار‌هایی بسیار مناسب برای سادهسازی برنامه نویسی اند. در بخش‌های آینده از کتابخانه‌های مختلف استفاده خواهیم کرد.

## Help

در بخش Help می‌توانید از منابعی که آردینو برای شما فراهم کرده است استفاده کنید. یکی از این منابع Reference است که در آن تمامی دستورها و تعاریف مربوط به فضای برنامه نویسی آردینو آمده است. این منبع به صورت پیش فرض در برنامه وجود دارد و می‌توانید بدون اتصال 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 خواهد شد.

### سریال مانیتور

گاهی 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 است تحت عنوان Serial Monitor در آردینو معرفی شده است. در بخش سریال ما به بررسی این ابزار قدرتمند خواهیم پرداخت.

### سریال پلاتر

در بعضی از موارد نیاز دارید داده‌هایی را که توسط سنسور‌ها می‌خوانید، تحلیل کنید. برای این کار راحت‌تر است داده‌ها به صورت نمودار نمایش داده ‌شوند تا تحلیل آنها ساده‌تر باشد. فرضا 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 با این دستور داده‌هایی را که از آردینو به رایانه خود می‌فرستید در واحد زمان به صورت نمودار نمایش دهید.

### برد

شرکت آردینو بردهای بسیار زیادی دارد. در بخش معرفی آردینو تعدادی از برد‌های اصلی آردینو و نحوه انتخاب آنها بیان شد. حال با توجه به گفته‌های قبلی، میکروکنترلرهای این بردها با یک دیگر فرق می‌کند. برای همین هنگام آپلود کردن کد به آنها، منطقا نباید از یک الگوریتم استفاده شود. برای همین قبل از آپلود کردن کد، 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 کنید 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 می‌خواهید این کد به‌خصوص را به کدام یک از این بردها بفرستید؟ برای اینکه این مشکل پیش نیاید شما باید آدرس USB را که برد مورد نظر به آن وصل شده است برای نرم افزار مشخص کنید. به صورت پیش فرض ویندوز آدرس هر یک از USB ها را تحت عنوان 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 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.



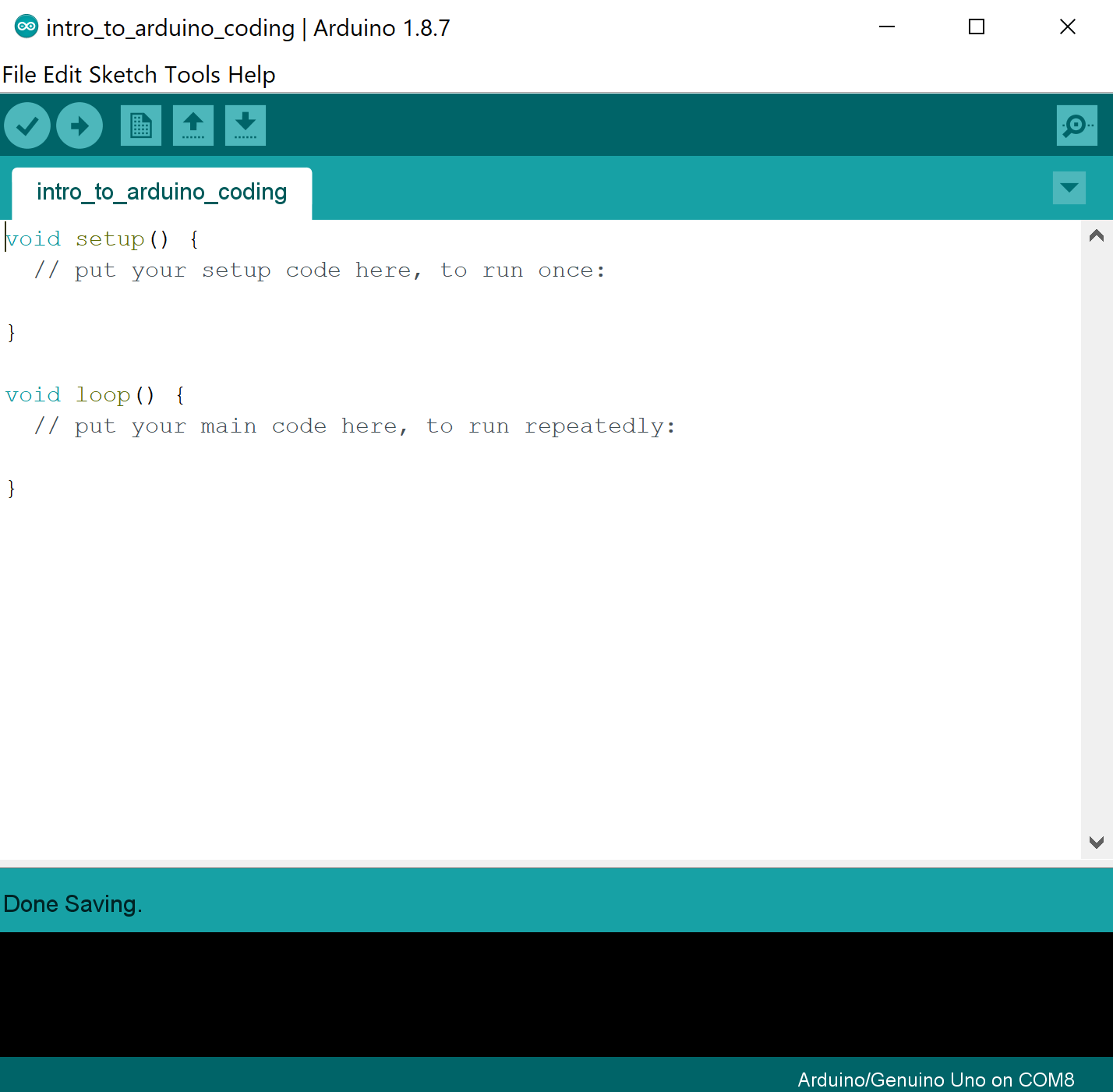
بسیار مناسب است که برای بالا رفتن سرعت یا از منوی دسترسی سریع استفاده کنید یا از shortcut های موجود در نرم افزار بهره ببرید. تمامی Shortcut ها به صورت زیر هستند:



# 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 کدهای ساده است.

در پروژه های کنترلی هدف همواره کنترل یک پارامتر است. به طور مثال می‌خواهید با یک فن، دمای هوای داخل یک محفظه را ثابت نگه دارید. برای این کار باید پیوسته دما را با یک سنسور دما 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 کنترلر اجرا کند نوشته می‌شود. همان طور که از اسم تابع مشخص است یک حلقه است که تا بینهایت اجرا می‌شود.

## void setup ( )

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

# یک برنامه ساده

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 با یکی از مثال‌های آردینو به نام چشمک زن این کار را انجام دادیم تا بتوانیم سالم بودن برد را بررسی کنیم. حال در اینجا می‌خواهیم خودمان این کد را بنویسیم. برای این کار ابتدا باید با دستورات زیر آشنا شویم.

## پین های دیجیتال

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

## pinMode ( )

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



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 است.

## digitalWrite ( )

این تابع 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 اند.



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 به واحد میلی ثانیه، برنامه را متوقف می‌کند. ورودی این تابع مقدار توقف به واحد میلی ثانیه است.



توجه: این تابع به صورت کامل برنامه را در این خط نگه می‌دارد؛یعنی، در این هنگام میکروکنترلر هیچ دستور دیگری را اجرا نخواهد کرد که این مسئله بسیار مهم است.

## پین دیود نوری داخلی

هر برد آردینو یک دیود نوری بر روی برد خود دارد. این دیود نوری به وسیله‌ی یک پین دیجیتال قابل دسترسی است. همان طور که در بخش معرفی آردینو گفته شد، مشخصات هر برد در سایت آردینو آمده است ودر آن شماره پینی که این دیود نوری به آن وصل شده است نیز ذکر شده است. گفتنی است که در اکثر بردهای 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 کدزنی حلقه‌ای دارد که 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 عکس‌برداری چشمتان پی ببرید.

توجه: متغیر led که در ابتدای برنامه تعریف شد، مقدار عددی 13 را در خود نگه می‌دارد. نکته قابل توجه این است 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 متغیرها مراجعه کنید.