# دیود نورانی چیست؟

دیود قطعه‌ای الکتریکی است که جریان را در یک جهت از خود عبور می‌دهد و در جهت دیگر در برار عبور جریان از خود مقاومت بالایی نشان می‌دهد. حال دیود نورانی نوعی دیود می‌باشد که اگر در جهت درست از آن جریان الکتریکی عبور داده شود، نور تولید می‌کند.

# تاریخچه‌ی دیود نورانی

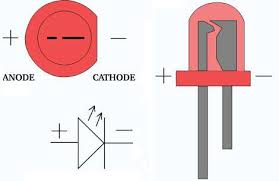
اولین دیود های نورانی در سال 196۲ میلادی و تنها با رنگ قرمز به صورت صنعتی تولید و وارد بازار شدند. دیود های نورانی سبز، آبی، زرد و نارنجی در دهه 70 میلادی تولید شدند. بهره نوری دیود های نورانی رفته رفته افزایش یافتند تا اینکه در دهه ۸0 و اوایل دهه 90 میلادی، آن ها به صورت گروهی و با کارایی بسیار بالا وارد بازار شدند. دیود های نورانی اولیه به علت بهره پایینشان، تنها در مدارات الکترونیکی استفاده می‌شدند، اما در حال حاضر همان طور که می‌دانید وارد مصارف خانگی شده‌اند و جایگاه لامپ های کم مصرف را گرفته‌اند.

# طیف نوری و انواع دیود نورانی

در واقع طیف نوری دیود های نوری تقریبا تمامی طیف نور را دربر می‌گیرد. این طیف شامل تمامی نور مرئی، مادون قرمز و فرابنفش می‌باشد. شدت نور تولیدی یک دیود نورانی به جریان آن بستگی دارد برای همین در بعضی از موارد برای راه اندازی دیود های نوری از منبع جریان استفاده می‌کنند. معمولا دیود های نوری از توان پایینی برخورداراند و بسیار کم مصرف هستند. این خواص باعث شده تا کاربرد این قطعه الکتریکی بسیار بالا باشد. در چراغ های راهنمایی رانندگی، علائم سطح شهر، چراغ های خودرو، روشنایی در موزه ها (به دلیل نداشتن پرتو ماورای بنفش برای اشیاء داخل موزه مضر نیستند) و در بسیاری از موارد دیگر کاربرد دارند.

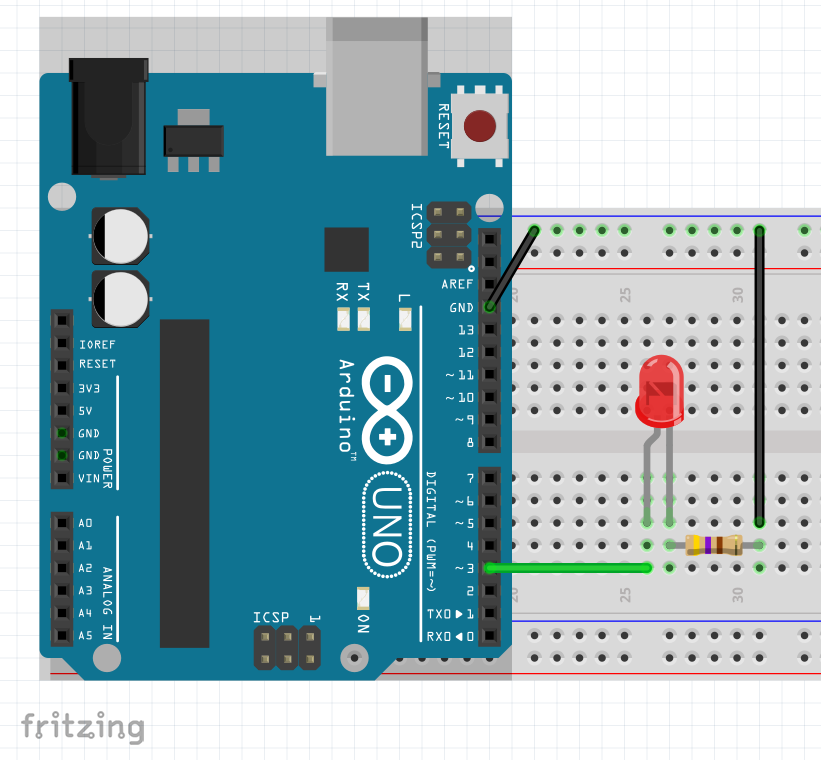
# تشخیص و نحوه عملکرد دیود

دیود نورانی همان طور در قبل گفته شد دیودی است که عبور جریان الکتریکی در جهت درست از داخل آن، باعث تولید نور می‌شود. پس در ابتدا باید بتوان جهت درست یک دیود نورانی یا همان پایه مثبت و منفی آن را به درستی تشخیص دهیم. برای این کار 2 راه وجود دارد. راه اول فقط مناسب برای دیود هایی است که قبل از آن طول پایه های آنها تغییر نکرده باشد. وقتی شما دیود را در ابتدا خریداری می‌کنید، یکی از پایه های آن بلند تر از دیگری می‌باشد. پایه بلند تر پایه مثبت می‌باشد. به بیان دیگر عبور جریان الکتریکی فقط از این پایه به پایه منفی ممکن است، اگر به صورت معکوس این کار انجام شود باعث شکسته شدن دیود می‌شود که درباره این موضوع به صورت کامل در بخش پیش نیاز های الکتریکی توضیح داده شده است. راه حل دوم نگاه کردن به داخل دیود است. اگر به شکل زیر توجه کنید می‌بینید که در داخل یک دیود نورانی 2 صفحه وجود دارد که یکی از دیگری کوچکتر است. صفحه‌ی کوچکتر نمایانگر پایه مثبت بوده و صفحه بزرگتر نمایانگر پایه منفی می‌باشد. این روش برای پیدا کردن جهت دیود هایی مناسب است پایه های آنها در گذشته بریده شده باشند. (مثلا برای لحیم کردن در داخل یک برد چاپی)



# مدار دیود نورانی

همان طور که در بخش معرفی آردینو گفته شد، برد های آردینو به هنگام نمایش مقدار 1 باینری ولتاژ 5 ولت را در خروجی خود تولید می‌کنند. حال شما فرض بفرمایید که ما می‌خواهیم توسط یکی از پین های آردینو یک ال ای دی را خاموش و روشن کنیم. این کار را می‌توانیم با صفر و یک کردن مقدار باینری آن پایه انجام بدهیم. برای این کار ابتدا باید مدار مورد نیاز را بدانیم. مدار راه اندازی ال ای دی بسیار ساده می‌باشد. برای راه اندازی یک ال ای دی فقط نیاز است که بعد از پایه منفی ال ای دی، مقاوتی قرار بدهیم تا جریان الکتریکی را محدود کند. برای این کار ما از مقاومت 470 اهمی استفاده می‌کنیم. این مقاومت با کد رنگی زرد-بنفش-قهوه‌ای کد گزاری می‌شود. بقیه مدار نیز مانند شکل بسته می‌شود. بعد از انجام بخش مربوط به مدارات الکتریکی، حال نوبت به آن رسیده که بخش مربوط به برنامه نویسی را انجام بدهیم که در این بخش یعنی قطع و وصل کردن پین شماره 3 برد آردینو. در این مدار اگر ولتاژ 5 ولت (1 باینری) به پین شماره 3 داده شود، ال ای دی روشن خواهد شد و اگر 0 (0 باینری)، ال ای دی خاموش خواهد شد.



# چشمک زدن دیود نوری بدون توقف

در این بخش دوباره می‌خواهیم دیود نوری را خاموش و روشن کنیم. در بخش معرفی فضای برنامه نویسی دیود نوری داخلی آردینو را که به صورت پیش فرض روی تمامی برد های آردینو می‌باشد را خاموش-روشن کردیم. در اینجا می‌خواهیم همان کار را انجام دهیم با این تفاوت که دیگر دیود نوری داخلی برد نیست بلکه مداری خارجی می‌باشد و دوم اینکه نمی‌خواهیم از تابع delay() استفاده کنیم. هم طور که گفته شد تابع delay تمامی برنامه را برای مدت زمان مشخصی متوقف می‌کند. این مشکل بزرگیست. در حالت واقعی شما همیشه از میکروکنترلر خود انتظار دارید که پارامتر هایی را بخواند و دستور های بدهد. اگر از تابع delay استفاده کنید در واقع باعث شدید که میکروکنترلر برای مدتی هیچ کاری انجام ندهد که این امر مشکل ساز می‌باشد. به طور مثال شما سیستم اتفاء حریقی ساخته‌اید. اگر این سیستم در لحظه‌ای که باید آب را وصل نکند ممکن است مشکلات بسیار زیادی به بار بیاید. برای همین استفاده از این تابع به جز در شرایط خاص مناسب نمی‌باشد و توصیه می‌شود از تابع دیگری به نام millis استفاده کنید.

## millis ( )

فراخوانی این تابع مدت زمانی را که میکروکنترلر در حال اجرا کردن برنامه فعلی بوده است را بر واحد میلی ثانیه بر می‌گرداند. عدد این تابع نیز بعد از 50 روز صفر می‌شود. به بیان ساده تر از ابتدای شروع برنامه شمارنده‌ای در حال کار است که هر شمارش آن زمان به خصوصی طول می‌کشد. آخرین عدد این شمارنده در جایی ذخیره شده که توسط این تابع، شما می‌توانید زمان مورد نیاز برای آن تعداد شمارش را دریافت کنید.



## abs ( )

این تابع مقدار قدر مطلق را بر می‌گرداند. به بیان ساده تر اگر عدد داده شده به تابع بزرگتر مساوی صفر باشد، خود آن عدد را بر می‌گرداند و اگر کمتر از صفر باشد، قرینه آن را بر می‌گرداند. به همین دلیل خروجی این تابع همواره مثبت می‌باشد.



## کد چشمک زن بدون توقف

باید توجه کرد که پایه مثبت دیود به پین شماره 3 وصل شده پس باید وضعیت این پین را تغییر دهیم تا دیود شروع به چشمک زدن کند. برای استفاده از تابع millis باید در دو زمان مختلف این تابع را فراخوانی کنیم و اگر اختلاف این 2 زمان بیشتر از مکث مورد نظر ما بود وضعیت دیود را تغییر دهیم.



## کد اصلاحی چشمک زن بدون توقف

توجه: کد قبلی در زمانی ممکن است دچار یک باگ شود. همان طور که در تعریف تابع millis گفته شد، این تابع بعد از 50 روز دوباره 0 می‌شود. اول اینکه به هنگام صفر شدن ممکن است متغیری که زمان اولیه را در خود ذخیره می‌کند قبل از صفر شدن زمان را ذخیره کند و سپس تابع صفر شود و بعد متغیر زمان ثانویه مقداردهی شود. اینگونه متغیر زمان اولیه بزرگتر از متغیر زمان ثانویه می‌باشد و تفاضل این دو منفی خواهد بود. این در حالی است که شرط ما بر روی "تفاضل بزرگتر مساوی صفر" گذاشته شده است. دوم اینکه یک متغیر int فقط می‌تواند تا عدد 32768 را در خود ذخیره کند ولی تابع millis تا 50 روز که معادل 4.3 میلیارد میلی ثانیه می‌باشد صفر نمی‌شود. به همین دلیل ساختار داده‌ای int نمی‌تواند این عدد را بعد از یک مدت در خود ذخیره کند (تابع millis خروجی‌اش یک متغیر unsigned long می‌باشد).

برای مشکل اول ما باید به جای استفاده از تفاضل دو عدد از اختلاف این 2 عدد یا همان قدر مطلق تفاضل آنها استفاده کنیم. برای این کار از تابع abs استفاده خواهیم کرد.

برای حل مشکل دوم 2 راه حل داریم. اول اینکه از داده هایی استفاده کنیم که فضای بیشتری از حافظه را اشغال می‌کند و توانایی ذخیره کردن اعداد بزرگتری را دارد. به طور مثال می‌توانیم از unsigned long استفاده کنیم. این راه حل راه حل مناسبی نیست چون خیلی راحت است. در واقع ما با این کار فضای حافظه مان را بی دلیل اشغال می‌کنیم و این کار بسیار نسنجیده‌ای است. راه بهتری وجود دارد. در بخش پیش نیاز برنامه نویسی به عملگری اشاره شد که خروجی آن باقی مانده 2 عدد بود (%). ما می‌توانیم باقی مانده تابع millis را بر 10000 پیدا کنیم و هر وقت این باقیمانده به اندازه 1000 واحد تغییر کرد عملیات تغییر وضعیت را انجام دهیم. البته این راه حل نیز مشکلی بزرگ دارد آن هم اینکه توان پردازشی قابل توجهی را به خاطر عملگر باقی مانده اشغال می‌کند (البته در مثال های ما تفاوت این 2 راه حل دیده نمی‌شود ولی در پروژه های بخصوصی شاید با این مشکلات روبه‌رو شوید). در آینده با عملگر های بیتی آشنا می‌شویم و می‌توانیم همین مثال را بدون توان پردازشی بالا حل کنیم.



## کوتاه ترین کد چشمک زن تا به حال

این کوتاه ترین کدی است که با معلوماتان تا به اینجای کار می‌توانید برای یک چشمک زن ساده بنویسید (حداقل ما تا به الان توانسته‌ایم). توضیحات این بخش با خودتان فقط این را بدانید که تابع digitalWrite هر متغیر باینری را قبول می‌کند. یعنی هر عدد بزرگتر از صفر HIGH خواهد بود و صفر LOW خواهد بود.



توجه: هیچ وقت کدتان را این قدر ساده نکنید. این کار باعث پیچیده شدن کد می‌شود و تغییر و اصلاح آن بسیار سخت می‌شود. همیشه سعی کنید کدی خوانا و ساده بنویسید که دنبال کردن آن کار ساده‌ای باشد.