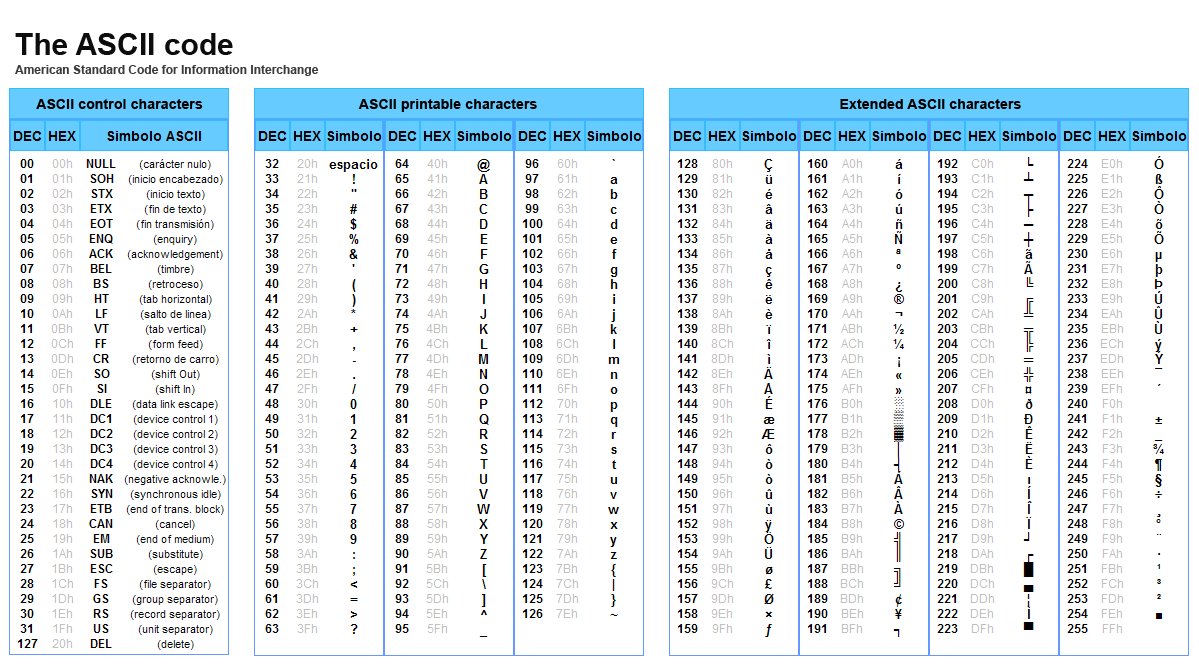
# سریال بخش دوم

در قسمت سریال بخش اول ابزار سریال معرفی شد. به وسیله‌ی این ابزار توانستیم اعداد و اطلاعاتی را که مورد نیازمان بود از برد آردینو دریافت کنیم و در صفحه نمایش لپتاپ نشان دهیم. در این بخش می‌خواهیم از طریق سریال مانیتور اطلاعاتی را به برد آردینو بفرستیم.

# خواندن اطلاعات فرستاده شده

اطلاعات در ارتباط سریالی بر اساس یک استاندارد فرستاده می‌شوند. داده ها در بسته‌های 1 بایتی (8 بیتی) فرستاده می‌شوند. هر داده یا هر کاراکتر در دنیای دیجیتال دارای یک نماد است و هر نماد توسط یک بایت کد گزاری شده است. تمامی این اطلاعات در جدولی به نام اسکی (ASCII) آورده شده است.



در این جدول هر کاراکتر یا دستور به عددی بین 0 تا 255 نسبت داده شده‌ است. در واقع برای فرستادن یک کاراکتر ابتدا کاراکتر را با عددی که در جدول اسکی به آن نسبت داده شده است جایگزین می‌شود. سپس آن عدد بر مبنای 2 برده می‌شود و در نهایت بر اساس روشی که در قسمت سریال بخش اول بیان شد، اطلاعات بیت به بیت فرستاده می‌شود.حال می‌خواهیم ببینیم که بعد از فرستاده شدن اطلاعات، چگونه این اطلاعات در سر دیگر ارتباط سریالی خوانده می‌شود؟

در واقع اطلاعات پشت سر هم ارسال می‌شوند و در buffer ذخیره می‌شوند. میکروکنترلر بعد از اینکه تعداد بایت مشخصی در بافر قرار گرفت، اطلاعات داخل آن را می‌خواند و دوباره منتظر می‌ماند تا در داخل بافر اطلاعات به اندازه کافی ذخیره شود تا دوباره عملیات خواند را اجرا کند. در حالت کلی شما می‌توانید مقدار حجم اطلاعات مورد نیاز برای اجرای عملیات خواند را تعیین کنید. به طور مثال می‌توانید به میکرو دستور بدهید بعد از هر 2 بایت اطلاعات را بخواند و یا بعد از هر 4 بایت. تفاوت این دو کار در این می‌باشد که هر چه حجم این اطلاعات کمتر باشد، میکرو در فاصله زمانی کمتری باید بافر را بخواند و این باعث تأخیر در برنامه خواهد شد. اسلاعات خوانده شده بر مبنای 2 می‌باشد برای همین آن ها را به عددی بر مبنای 10 می‌بریم و سپس بر اساس جدول اسکی کاراکتر به خصوص را جایگزین عدد می‌کنیم.

نکته: تمامی اطلاعاتی که توسط ارتباط سریالی خوانده می‌شود به صورت متغیر string ذخیره خواهد شد و برای استفاده از این اطلاعات، باید آن ها را به فرم مناسب در بیاوریم. به طور مثال اگر شما عدد 2 را از طریق سریال بفرستید در طرف دیگر ارتباط کاراکتر 2 دریافت خواهد شد. شما باید به نحوی این کاراکتر را تبدیل به عدد 2 بکنید و سپس از این عدد در عملیات های ریاضی استفاده کنید.

# تبدیل اطلاعات به داده های مفید

در این بخش می‌خواهیم یک کاراکتر را از طریق سریال مانیتور به آردینو ارسال کنیم و سپس از برد آردینو بازخورد مناسب را دریافت کنیم. ابتدا باید با دستورات زیر آشنا شویم:

## Serial.read()

این تابع اولین کاراکتر (اولین بایت) از داده های ارسال شده از پورت سریال به برد را می‌خواند. به طور مثال اگر مجموعه کاراکتر ABC از طریق سریال به برد ارسال شود، با این تابع شما می‌توانید کارکتر A را بخوانید. سپس با دوباره اجرا کردن این دستور کارکتر B را بخوانید و به همین ترتیب جلو بروید.



### Serial.available()

این تابع وجود اطلاعات در پورت سریال را بررسی می‌کند. به بیان ساده تر اگر اطلاعاتی در حال دریافت شدن ( آماده برای خوانده شدن) در پورت سریال موجود باشد، خروجی این تابع True یا 1 باینری خواهد بود. دقیق تر اگر بخواهیم بگوییم این تابع تغییر وضعیت پورت سریال را از حالت 0 به 1 بررسی می‌کند و در صورت وجود، آن را گزارش می‌کند.



### تعریف کردن یک تابع

در آردینو برای اینکه یک تابع را تعریف کنیم مانند C++ عمل می‌کنیم. باید نوع تابع. و ورودی تابع را مشخص ‌کنیم.

در آردینو 2 نوع تابع وجود دارد، توابعی که خروجی دارند و توابعی که خروجی ندارند. توابعی که خروجی دارند از نوع int خواهند بود و توابعی که خروجی ندارند از نوع void هستند.

ورودی توابع نیز می‌تواند از انواع متغیر ها باشد. به طور مثال اگر ورودی تابع یک عدد صحیح است، ورودی تابع را از نوع int می‌گذاریم.



# روشن کردن ال ای دی به وسیله پورت سریال

در این بخش می‌خواهیم به وسیله فرستادن یک کاراکتر خاص ال ای دی را روشن کنیم و با ارسال کاراکتری دیگر ال ای دی را خاموش کنیم. در اینجا می‌خواهیم با ارسال کاراکتر O ال ای دی را روشن کنیم و با ارسال کاراکتر F ال ای دی را خاموش کنیم. ال ای دی به دیجیتال پین 5 وصل شده است. برای این کار ابتدا به وسیله یک تابع ورودی پورت سریال را دریافت می‌کنیم. سپس ورودی را با کاراکتر های O و F مقایسه می‌کنیم و در نهایت دستور درست را اجرا می‌کنیم.



حال کمی کد را عوض کنیم. این بار می‌خواهیم تا وقتی کاربر کلید O را نگه می‌دارد دیود نوری روشن بماند و هر موقع کاربر کلید O را رها کرد و یا کلید دیگری را فشار داد به اشتباه، دیود نوری خاموش شود. برای این کار باید ما یک نوع ساده از توابع وقفه را تولید کنیم که به صورت کامل در بخش شافت انکودر ها توضیح داده شده است. وظیفه این نوع توابع بررسی کردن تغییرات وضعیت دیجیتالی یک پین مشخص در طول زمان می‌باشد و اگر تغییر وضعیتی را ثبت کنند عملیات از پیش تعریف شده‌ای را انجام دهند. به طور مثال در این مثال ما باید وضعیت ارسال شدن حرف O را مد نظر بگیریم. در صورت فرستاده نشدن این حرف و یا فرستاده شدن حرف دیگر ما باید دیود را خاموش کنیم. لازم به ذکر است که توابع وقفه بدون اینکه توان پردازشی اصلی میکرو را درگیر کنند به صورت دائم در پشت صحنه اجرا می‌شوند. ما برای اینکه بتوان وضعیت ارسال شدن حرف O را مانیتور کنیم از یک تابع استفاده می‌کنیم که در آن به صورت پیوسته وضعیت پورت را بررسی می‌کنیم.



توجه: در حالت کلی شما نمی‌توانید با نگه داشتن کلید O بر روی کیبورد خود همزمان حرف O را برای بورد آردینو ارسال کنید و باید کلید Enter را فشار دهید. پس در حقیقت کلید فشاری وجود ندارد ولی می‌توانید به وسیله ارسال کردن چندین O به صورت یک پکیج داده، وضعیت کلید فشاری را شبیه سازی کنید. دلیل نوشتن تابع دیلی در خط آخر به خاطر همین موضوع می‌باشد چون زمان پردازش کردن 100 تا حرف O پشت سر هم بسیار کم بوده و قابل روئت نیست برای همین با کمی توقف درستی کد سنجیده می‌شود. حال اگر می‌خواهید به صورت کاملا درست یک کلید فشاری درست کنید باید از سخت افزار های خارجی مانند یک کلید و یا کی‌پد استفاده کنید.