فهرست

مقدمه ۱

فصل ۰. مفاهیم پایه ۳

قطعات مورد نیاز برای انجام پروژه‌ها ۳

آشنایی با مفاهیم مقدماتی ۵

نمادهای الکتریکی پایه ۵

مفهوم ولتاژ و جریان الکتریکی ۷

مفهوم سری و موازی ۸

قطعات الکترونیکی پُرکاربرد ۹

بِردبورد ۹

تَک‌سوییچ ۱۰

مقاومت ۱۱

پتانسیومتر ۱۳

خازن ۱۴

القاگر ۱۶

نوسان‌ساز کریستالی ۱۷

دیود ۱۸

ال‌ای‌دی ۱۹

سون‌سگمنت ۱۹

ترانزیستور ۲۱

مدارات مجتمع (آی‌سی) ۲۳

پروژه‌های الکترونیک ۲۴

پروژه‌ی اوّل: مقاومت و ال‌ای‌دی ۲۴

پروژه‌ی دوم: مقاومت، ال‌ای‌دی و کلید فشاری ۲۷

پروژه‌ی سوم: مقاومت، ال‌ای‌دی و حسگر شیب ۲۹

پروژه‌ی چهارم: مقاومت، ال‌ای‌دی و پتانسیومتر ۳۱
پروژه‌ی پنجم: مقاومت، ال‌ای‌دی و حسگر نور ۳۳
پروژه‌ی ششم: مقاومت، ال‌ای‌دی و دیود ۳۵
پروژه‌ی هفتم: مقاومت، ال‌ای‌دی، دیود و خازن ۳۷
پروژه‌ی هشتم: مقاومت، ال‌ای‌دی و ترانزیستور ۴۰
پروژه‌ی نهم: مقاومت، ال‌ای‌دی، القاگر و ترانزیستور ۴۲
پروژه‌ی دهم: ال‌ای‌دی، مقاومت، خازن و آی‌سی ۴۴
پروژه‌ی یازدهم: کلید فشاری، ال‌ای‌دی، مقاومت و آی‌سی ۴۸
پروژه‌ی دوازدهم: کلید فشاری، ال‌ای‌دی، مقاومت، خازن و آی‌سی ۵۰

فصل ۱. معرفی آردوینو ۵۳
آشنایی با آردوینو ۵۳
تاریخچه ۵۵
آردوینو یا رسپبری‌پای؟ ۵۷
مثال‌های کارایی و عدم‌کارایی آردوینو ۵۹

فصل ۲. سخت‌افزار آردوینو ۶۳
تقسیم وظایف و بخش‌بندی کلی سخت‌افزار ۶۵
چیدمان اجزاء و قطعات ۶۶
۱. میکروکنترلر ۷۱
۲. بخش تغذیه ۷۷
۳. درگاه ارتباط با رایانه ۸۰
۴. درگاه‌های گروهی ۸۲
سیگنال‌های آنالوگ و دیجیتال ۸۲
درگاه‌های گروهی ۸۵
گروه الف. درگاه‌های قدرت: ۸۶
گروه ب. درگاه‌های آنالوگ: ۸۸
گروه ج. درگاه‌های دیجیتال: ۸۸

فصل ۳. انواع بوردهای آردوینو ۹۱
آردوینو اونو ۹۲
آردوینو نانو ۹۳
آردوینو مگا ۹۴
آردوینو پرو ۹۵

آردوینو پرو مینی ۹۶
آردوینو لئوناردو ۹۷
آردوینو دوئه ۹۸
آردوینو میکرو ۹۹
آردوینو مینی ۱۰۰
آردوینو زیرو ۱۰۱
آردوینو لیلی‌پَد ۱۰۲
آردوینو یئون ۱۰۳
آردوینو اِم‌کی‌آر ۱۰۴
آردوینو ۱۰۱ ۱۰۵
جدول تطبیقی بوردهای آردوینو ۱۰۶
فصل ۴. ماژول‌ها و شیلدها ۱۰۷
تعریف قطعه، ماژول و شیلد ۱۰۷
حسگرها و عملگرها ۱۰۹
حسگر دما ۱۱۰
حسگر دما و رطوبت ۱۱۱
حسگر نور ۱۱۲
مولد مادون‌قرمز ۱۱۳
حسگر مادون‌قرمز ۱۱۳
مولد لیزر ۱۱۴
مولد صدا ۱۱۴
حسگر صدا ۱۱۶
حسگر تشخیص گاز و دود ۱۱۶
حسگر رطوبت خاک ۱۲۰
حسگر قطرات باران ۱۲۰
ال‌ای‌دی دو رنگ ۱۲۱
ال‌ای‌دی تمام‌رنگی ۱۲۲
حسگر تشخیص رنگ ۱۲۳
حسگر شیب ۱۲۴
حسگر لرزش ۱۲۵
حسگر فشار ۱۲۶
حسگر خمش ۱۲۷
حسگر شتاب ۱۲۸

ژیروسکوپ ۱۲۸
کلید فشاری ۱۲۹
کلید چرخشی رمزگذار ۱۳۰
دسته‌فرمان (جوی استیک) ۱۳۰
حسگر مغناطیسی زبانه ای ۱۳۱
حسگر مغناطیسی اثر هال ۱۳۲
حسگر لمس خازنی ۱۳۳
حسگر قطع نور ۱۳۴
حسگر شعله ۱۳۵
حسگر تشخیص حرکت ۱۳۶
حسگر فاصله‌یاب فراصوتی ۱۳۷
ماژول آراف‌آی‌دی ۱۳۸
ماژول ماتریس ال‌ای‌دی ۱۳۸
ماژول ماتریس ال‌سی‌دی ۱۳۹
شیلد کنترل موتور ۱۴۰
الف. موتور الکتریکی ساده ۱۴۰
ب. سروو موتور ۱۴۰
ج. استپر موتور ۱۴۱
ماژول رله ۱۴۲
ماژول ارتباط با اینترنت ۱۴۴
فصل ۵. نرم‌افزار آردوینو ۱۴۵
نصب نرم‌افزار آردوینو ۱۴۵
آشنایی با محیط نرم‌افزار آردوینو ۱۴۸
نوار فهرست و زیرشاخه‌های آن ۱۵۰
برنامه‌نویسی در محیط آردوینو ۱۵۵
آشنایی با علایم ۱۵۵
ثابت‌ها و متغیرها ۱۵۷
انتساب مقدار به متغیر ۱۶۰
آرایه‌ها ۱۶۲
رشته‌ها ۱۶۳
تابع‌ها ۱۶۴
setup() و loop() ۱۶۶

عبارات توضیحی ۱۶۹
خروج داده‌ها از نمایشگر سریال ۱۷۲
خروج داده‌ها از رسّام‌سریال ۱۷۷
کنترل روشن و خاموش‌شدن ال‌ای‌دی ۱۷۸
pinMode() ۱۷۹
digitalWrite() ۱۸۰
delay() ۱۸۱
delayMicroseconds() ۱۸۳
ورود داده‌ها به نمایشگر سریال ۱۸۴
if ۱۸۶
if...else ۱۸۸
for ۱۹۰
while ۱۹۳
do...while ۱۹۴
switch...case ۱۹۵
break ۱۹۶
continue ۱۹۷
return ۱۹۸
digitalRead() ۲۰۱
analogRead() ۲۰۳
analogReference() ۲۰۴
analogWrite() ۲۰۵
millis() ۲۰۷
micros() ۲۰۸
tone() ۲۰۹
noTone() ۲۱۰
pulseIn() ۲۱۱
const ۲۱۳
#define ۲۱۵
random() ۲۱۶
عملگرهای ریاضی ۲۱۶
abs() ۲۱۶
constrain() ۲۱۷

map() ۲۱۷
max() ۲۱۸
min() ۲۱۹
pow() ۲۱۹
sq() ۲۱۹
sqrt() ۲۲۰
عملگرهای مثلثاتی ۲۲۰
sin() ۲۲۰
cos() ۲۲۰
tan() ۲۲۱
تجزیه و تحلیل نویسه (کاراکتر) ۲۲۱
مبدّل نوع داده‌ها ۲۲۳
کتابخانه‌ها ۲۲۳
کتابخانه‌های استاندارد آردوینو ۲۲۶
کتابخانه‌های اضافی آردوینو ۲۲۶

فصل ۶. پروژه‌های آردوینو ۲۲۹
پروژه‌ی اول: روشن و خاموش‌کردن ال‌ای‌دی توسط آردوینو ۲۲۹
پروژه‌ی دوم: تغییر میزان روشنایی ال‌ای‌دی توسط آردوینو ۲۳۳
پروژه‌ی سوم: کنترل ال‌ای‌دی توسط کلید فشاری و آردوینو ۲۳۶
پروژه‌ی چهارم: کنترل ال‌ای‌دی توسط دو کلید فشاری و آردوینو ۲۴۲
پروژه‌ی پنجم: کنترل ال‌ای‌دی توسط پتانسیومتر و آردوینو ۲۴۵
پروژه‌ی ششم: کنترل ال‌ای‌دی توسط حسگر نور ۲۴۹
پروژه‌ی هفتم: کنترل سروو موتور توسط آردوینو ۲۵۳
پروژه‌ی هشتم: کنترل سون‌سگمنت توسط آردوینو ۲۵۷
پروژه‌ی نهم: حسگر فاصله‌یاب فراصوتی ۲۶۲
پروژه‌ی دهم : کلید صوتی و رله ۲۶۷
پروژه‌ی یازدهم: ماژول ارتباط با اینترنت ۲۷۳
الف. حالت برنامه‌دهی ۲۷۵
ب. حالت معمول اجرای برنامه ۲۷۸
تنظیم نرم‌افزار آردوینو برای ارتباط با ماژول ۲۷۸
ارتباط ماژول با مودم اینترنت ۲۷۹

کنترل ال‌ای‌دی از طریق اینترنت ۲۸۱
مدار تثبیت‌کننده‌ی ولتاژ برای ماژول ۲۸۴
پروژه‌ی دوازدهم: فشرده‌سازی آردوینو در یک آی‌سی ۲۸۶
منابع ۲۹۱

مقدمه

امروزه با پیشرفت صنایع الکترونیک و توسعه‌ی روزافزون حسگرها و عملگرها از یک سو و تولید انبوه و کاهش قیمت آن‌ها از سوی دیگر، زمینه‌ی کاربری‌شان در محصولات ساده و روزمره، بیش از پیش فراهم شده است. هوشمندسازی ساختمان‌ها، تعامل‌گرایی محصولات، و ارتباط اشیاء با شبکه‌های رایانه‌ای از دستاوردهای توسعه‌ی فناوری الکترونیک است که در زندگی روزمره تداعی یافته است. هم‌سو با توسعه‌ی زیرساخت‌های سخت‌افزاری، سهولت کاربری نرم‌افزارها نیز افزایش یافته است. در این میان، آردوینو به‌عنوان بستری مشتمل بر سخت‌افزار و نرم‌افزار منبع‌باز و کاربرپسند، جایگاه خود را نه تنها در میان دانشجویان علوم مهندسی، بلکه در بین طراحان و علاقه‌مندان به علوم تعاملی و توسعه‌ی محصول نیز به دست آورده است. سهولت کاربری و قیمت مناسب آردوینو، منجر به توسعه‌ی روزافزون جامعه‌ی کاربران آن در سطح جهان گردیده که این تعداد رو به فزونی کاربران نیز، خود از امتیازات آن محسوب می‌شود چرا که کمک و هم‌فکری سایر افراد، چالش اجرای پروژه‌های جدید را به حداقل می‌رساند. کتاب حاضر، با بهره‌گیری از معتبرترین منابع آموزشی آردوینو در سطح جهانی تألیف گردیده و ضمن تشریح اصول استفاده از این بورد الکترونیکی، به ارایه‌ی مثال‌ها و پروژه‌های هدف‌مند و آزمایش‌شده می‌پردازد. به همین منظور، در این کتاب، پس از معرفی کلی آردوینو، به تشریح سخت‌افزار و نرم‌افزار آن پرداخته شده و ضمن معرفی اجمالی نمونه‌های متداول حسگرها و عملگرها، نحوه‌ی کاربرد برخی از آن‌ها در قالب پروژه‌های ساده توضیح داده می‌شود. هرچند هدف از نگارش این کتاب، آموزش الکترونیک و برنامه نویسی نیست ولی در بخش‌هایی که لازم است، توضیح مختصری در این خصوص ارایه می‌گردد. امید است که مورد استفاده‌ی مخاطبان قرارگیرد.

فصل ۱. معرفی آردوینو

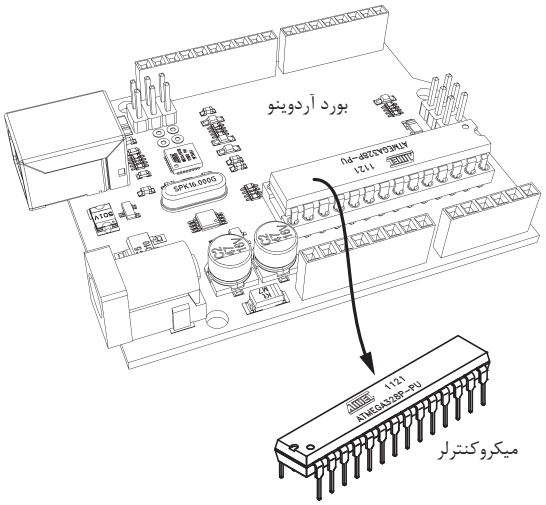
آشنایی با آردوینو

آردوینو[۱] سیستمی است که برای هوشمندسازی اشیاء فیزیکی و نیز ایجاد ارتباط بین این اشیاء با سایر اشیاء و یا انسان به کار می‌رود. آردوینو از دو بخش سخت‌افزاری و نرم‌افزاری تشکیل می‌شود. سخت‌افزار آردوینو یک بورد[۲] الکترونیکی نسبتاً ساده است که قابلیت ذخیره و اجرای دستورات و برنامه‌هایی را دارد که توسط نرم‌افزار آردوینو برای آن تعریف می‌شوند. برای این منظور، ابتدا نرم‌افزار آردوینو بر روی رایانه نصب می‌شود و سپس از طریق یک سیم رابط، دستورات برنامه‌نویسی شده در محیط نرم‌افزار، به بورد آردوینو منتقل و ذخیره می‌گردند. از این مرحله به بعد، دیگر نیازی به اتصال به رایانه نیست و بورد آردوینو می‌تواند این دستورات ذخیره شده را به دفعات نامحدود تکرار کند. کار کردن با آردوینو ساده است و نیازی به دانش حرفه‌ای الکترونیک و برنامه‌نویسی پیچیده ندارد. این امتیاز باعث شده که سیستم آردوینو به‌صورت گسترده‌ای مورد استفاده و توجه طراحان صنعتی، طراحان گرافیک و معماران قرارگیرد تا بتوانند با تکیه بر قابلت‌های آردوینو، سیستم‌هایی هوشمند را طراحی نموده و سطح تعامل محصول و محیط را با مخاطب افزایش دهند.

هم نرم‌افزار و هم سخت‌افزار آردوینو، متن‌باز[۳] هستند. نرم‌افزار آردوینو رایگان است و سخت‌افزار آن ارزان قیمت است. این سخت‌افزار نسبت به خطاهای معمول و متداول کاربران، مقاوم است. تعداد افرادی که در سطح جهان از آردوینو استفاده می‌کنند بسیار زیاد است و این امتیاز سبب سهولت در حل مشکلات بالقوه در پروژه‌های جدید کاربران می‌شود. در شکل ۱.۱ نمونه ای از بورد آردوینو نشان

۱. Arduino
۲. Board
۳. Open-source

داده شـده اسـت.



شکل ۱.۱: نمونه‌ای از بورد آردوینو و میکروکنترلر آن

بخـش اصلـی یـک بـورد آردوینـو، میکروکنترلر[۱] آن اسـت. امـروزه میکروکنترلرهـا در طیف وسـیعی از لـوازم و تجهیـزات روزمـره بـه کار می‌رونـد، مثـلاً کنتـرل تلویزیون، ماشـین‌های اداری، اسـباب‌بازی‌ها و محصـولات هوشـمند. میکروکنترلـر آردوینـو در بخـش مرکـزی بـورد الکترونیکـی نصـب شـده اسـت و سـایر اجـزاء و قطعـات در اطـراف آن قرارگرفتـه انـد (شـکل ۱.۱). دسـتورات، فرامیـن و برنامه‌هـا از طریـق رایانـه و بـا اسـتفاده از سـیم رابـط، بـر روی میکروکنترلـر آردوینـو ذخیـره می‌شـوند. همچنیـن دریافـت اطلاعـات ورودی از حسـگرها[۲] و ارسـال اطلاعـات خروجـی بـه عملگرهـا[۳] از طریـق پایه‌هـای ورودی و خروجـی میکروکنترلـر انجـام می‌شـود. جهـت سـهولت اتصـال سـیم‌های ارتباطـی، پایه‌هـای میکروکنترلـر بـه درگاه‌هـای حاشـیه‌ی بـورد آردوینـو متصـل شـده‌اند. ایـن درگاه‌هـا کامـلاً در دسـترس بـوده و بـه سـهولت می‌تـوان سـر سـیم‌های رابـط خارجـی را بـه داخـل آن فشـار داد و بـا میکروکنترلـر ارتبـاط برقـرار نمـود. همچنین تغذیـه الکتریکی میکروکنترلـر از طریـق بـورد آردوینـو انجـام می‌شـود. در واقـع نقـش بـورد آردوینـو آن

۱. Microcontroller
۲. Sensors
۳. Actuators

است که میکروکنترلر را تغذیه نموده و دسترسی به درگاه‌های ورودی و خروجی آن را آسان‌تر نماید. در صورت قطع شدن جریان تغذیه و یا جدا نمودن میکروکنترلر از بورد آردوینو، آسیبی به اطلاعات ذخیره شده بر روی آن نمی‌رسد. در مجموع، نقش بورد آردوینو آن است که سهولت استفاده و کاربر پسندی[1] میکروکنترلر را افزایش دهد.

به منظور درک بهتر کاربری آردوینو، مثالی را مطرح می‌کنیم: می‌خواهیم با استفاده از یک حسگر دما و یک بخاری برقی، دمای اتاق را در زمستان در محدوده‌ی ۲۰ درجه سانتیگراد نگه داریم. حسگر دما، اطلاعات ورودی را برای آردوینو تأمین می‌کند و بخاری برقی، فرامین خروجی را از آردوینو دریافت می‌نماید. در این مثال، آردوینو دو وظیفه را بر عهده دارد:

الف. اگر دمای اتاق به بالاتر از ۲۰ درجه‌ی سانتیگراد رسید، بخاری برقی را خاموش کند.

ب. اگر دمای اتاق به پایین‌تر از ۲۰ درجه‌ی سانتیگراد رسید، بخاری برقی را روشن کند.

با توجه به اینکه سخت‌افزار آردوینو به خودی خود نمی‌تواند تصمیم بگیرد که چه دمایی برای اتاق مناسب است بنابراین باید محدوده‌ی دمای مورد نظر خود را از طریق نرم‌افزار آردوینو برای آن تعیین کنیم. نکته‌ی حایزاهمیت آن است که کافی است تنها یک مرتبه دمای مورد نظر برای آردوینو تعریف شود تا بتواند به دفعات نامحدود، با خاموش و روشن‌کردن بخاری، آن دما را حفظ نماید.

تاریخچه

ایده‌ی ساخت آردوینو در سال ۲۰۰۳ میلادی در انستیتو طراحی‌تعاملی ایورئا[2] در کشور ایتالیا شکل گرفت: ایده عبارت بود از ساخت وسیله‌ای ساده و کم‌هزینه برای انجام پروژه‌های دیجیتال دانشجویان، به‌خصوص آن‌هایی که آشنایی چندانی با اصول مهندسی و برنامه‌نویسی ندارند. سه فرد کلیدی در به ثمر نشاندن این ایده نقش داشتند:

۱. User friendly

۲. Interaction Design Institute Ivrea

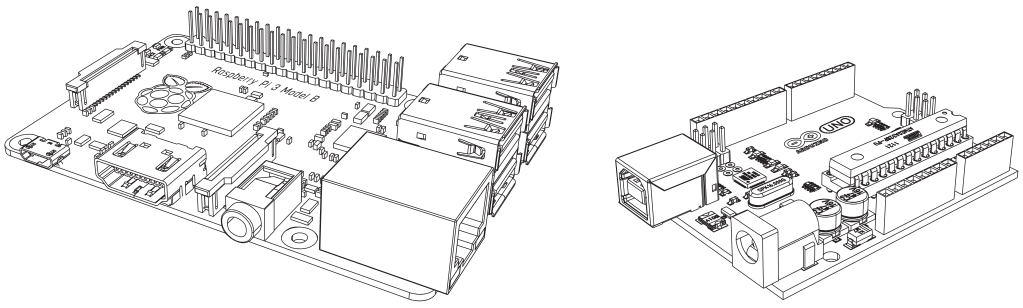
هرناندو باراگان[۱]، ماسیمو بانزی[۲]، و کیسی ریس[۳].
باراگان یکی از دانشجویان انستیتو ایورئا بود که تصمیم گرفت پایان‌نامه‌ی کارشناسی‌ارشد خود را در این زمینه اجراء نماید. بانزی و ریس نیز اساتید راهنمای پایان‌نامه‌ی باراگان بودند. تا آن زمان هنوز اسمی از آردوینو در میان نبود. نتیجه‌ی پایان‌نامه‌ی باراگان بسیار موفقیت‌آمیز بود و منجر به ایجاد سخت‌افزار و نرم‌افزاری شد که وایرینگ[۴] نام گرفت. سخت‌افزار وایرینگ ویژگی‌های مورد نظر را نسبت به سایر نمونه‌های موجود در بازار آن زمان داشت یعنی ساده و کم‌هزینه بود. نرم‌افزار وایرینگ نیز بر مبنای یکی از زبان‌های برنامه‌نویسی موجود به نام پروسسینگ[۵] تهیه شده بود.
پس از اتمام پایان‌نامه، بانزی درصدد کاهش هزینه‌های سخت‌افزار وایرینگ برآمد و در سال ۲۰۰۵ میلادی با همکاری دیوید کوآرتلس[۶] و دیوید ملیس[۷] (که به‌ترتیب کارمند و دانشجوی انستیتو ایورئا بودند)، به توسعه‌ی پروژه‌ی وایرینگ پرداخت و نام آن را به آردوینو تغییر داد. این نام جدید برگرفته از نام کافه‌ای به نام آردوین[۸] در شهر ایورئا بود که اکثر جلسات گروه در آنجا تشکیل می‌شد. واژه‌ی آردوین، نام یکی از شاهزاده‌گان قدیم ایتالیا است که زمانی حکمران شهر ایورئا بود و در قرن یازدهم میلادی به پادشاهی ایتالیا رسید.
رفته‌رفته هسته‌ی اصلی تیم آردوینو با حضور پنج نفر اصلی شکل گرفت: بانزی، کوآرتلس، ملیس، و دو فرد جدید به نام‌های تام ایگو[۹] و جیانلوکا مارتینو[۱۰]. در بین اعضای این گروه پنج نفره، نامی از باراگان (نویسنده‌ی پایان‌نامه) به چشم نمی‌خورد و هیچ‌گاه برای مشارکت در این گروه از وی دعوت نشد. این تیم پنج نفره در سال ۲۰۰۸ میلادی نام تجاری آردوینو را در آمریکا ثبت نمودند. در همین زمان، مارتینو، یکی از افراد همین گروه، به‌صورت پنهانی، نام آردوینو را برای خود در کشور ایتالیا ثبت کرد و به‌صورت موازی با شرکت اصلی، شروع به بهره‌برداری

۱. Hernando Barragán
۲. Massimo Banzi
۳. Casey Reas
۴. Wiring
۵. Processing
۶. David Cuartielles
۷. David Mellis
۸. Arduin
۹. Tom Igoe
۱۰. Gianluca Martino

شخصی از تولید و فروش این محصول نمود. پیچیدگی‌های حقوقی شکل گرفته پیرامون نام تجاری آردوینو، سبب شد که شرکت اصلی در آمریکا مجبور شود که برای عرضه‌ی محصولات خود در خارج از آمریکا، از نام تجاری گنوئینو[۱] استفاده نماید. مشخصات و ویژگی‌های بوردهای گنوئینو دقیقاً مشابه آردوینو است و هیچ تفاوتی با آن ندارد.

آردوینو یا رسپبری‌پای؟

امروزه انواع مختلفی از بوردهای الکترونیکی در بازار وجود دارند که به میکروکنترلرهای قابل برنامه‌نویسی مجهز اند. بخشی از آن‌ها کاربری حرفه‌ای و صنعتی دارند و برخی دیگر برای جنبه‌های آموزشی عرضه گردیده‌اند. همچنین اینترنت اشیاء، هوشمندسازی و طراحی‌تعاملی نیز از سرفصل‌های جدید و پرطرفدار در عرصه‌ی میکروکنترلرها می‌باشند. این گونه سخت‌افزارها، در طیف وسیعی از مدل‌ها و نام‌های تجاری به بازار عرضه می‌شوند. از نمونه‌های بیشتر شناخته شده در بازار ایران، آردوینو و رسپبری‌پای (رزبری‌پای)[۲] می‌باشند (شکل ۱.۲). سوالی که در وهله‌ی نخست مطرح می‌شود آن است که کدام یک از این دو نوع سخت‌افزار پاسخ‌گوی نیاز ما است؟ تفاوت این دو نوع با یکدیگر چیست و اصولاً هریک چه ویژگی‌هایی دارند؟



شکل ۱.۲: نمونه‌ای از بورد آردوینو (راست) و رسپبری‌پای (چپ)

۱. Genuino
۲. Raspberry Pi

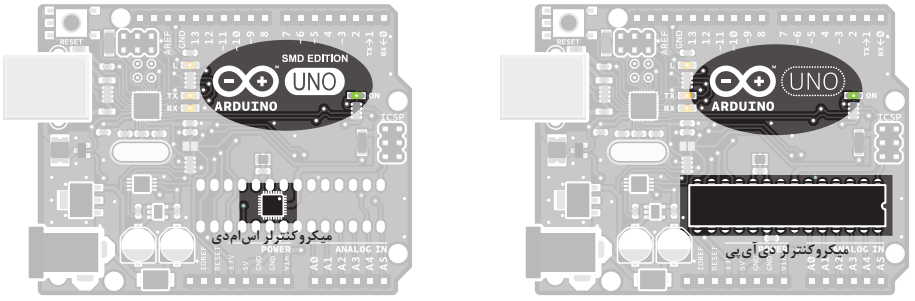
لرزش‌های مداوم و شدید، وجود مواد خورنده، و رطوبت محیطی، از جمله عواملی هستند که می‌توانند عملکرد سخت‌افزار آردوینو را با چالش‌های جدی مواجه کنند. به منظور تصمیم‌گیری صحیح در موارد کارایی و عدم‌کارایی آردوینو در پروژه‌های مختلف، به ذکر مثال‌هایی می‌پردازیم که در جدول ۱.۱ آمده اند.

جدول ۱.۱: مثال‌هایی از موارد کارایی و عدم‌کارایی آردوینو

مثال	کارایی آردوینو	توضیح
اندازه‌گیری دمای داخل خودرو	بله	-
حسگر فاصله تا مانع برای دنده‌عقب خودرو	بله	-
تصمیم‌گیری و فرمان به ترمزهای خودرو	خیر	نیاز به ایمنی بالا
تجهیزات اتاق‌عمل و بخش مراقبت‌های‌ویژه	خیر	نیاز به ایمنی بالا
حسگر ورود دست کارگر به دستگاه پرس	خیر	نیاز به ایمنی بالا
سیستم دزدگیر بانک	خیر	نیاز به امنیت بالا
پایش درگاه‌های فرودگاهی	خیر	نیاز به امنیت بالا
کارخانه‌ی تولید مواد شیمیایی	خیر	وجود مواد خورنده
قایق، کشتی و تجهیزات دریایی	خیر	رطوبت بسیار زیاد
پروژه‌های پایش کویر	خیر	گرد و غبار زیاد
دستگاه‌های غربال‌گری و خردکن صنعتی	خیر	لرزش زیاد
اتوماسیون صنعتی	خیر	نیاز به درگاه‌های زیاد
تغذیه، نور و دمای مرغداری و دامپروری	بله	-
تنظیم دما، رطوبت و آبیاری گلخانه	بله	-

فصل ۲. سخت‌افزار آردوینو

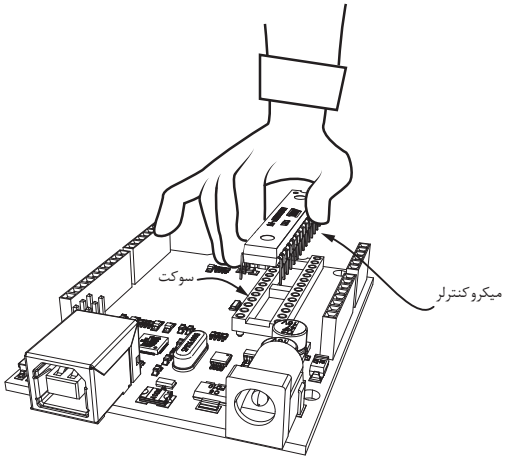
همان‌طور که در فصل قبل اشاره شد، بستر آردوینو دارای دو بخش سخت‌افزاری و نرم‌افزاری است. در این فصل به معرفی سخت‌افزار آردوینو پرداخته می‌شود. نمونه‌های مختلفی از بوردهای آردوینو در بازار وجود دارند ولی مناسب‌ترین نوع آن برای شروع یادگیری، مدل «اونو، آر۳، دی‌آی‌پی»[۱] است که توضیحات این بخش از کتاب، منطبق با آن ارایه می‌شوند (شکل ۲.۱). واژه‌ی «اونو»[۲] در زبان ایتالیایی به‌معنای «عدد یک» است. «آر۳»[۳] یعنی سری سوم از بورد آردوینوی اونو، که نسبت به نمونه‌های قبلی، مورد بازنگری و ارتقاء قرارگرفته است. عبارت «دی‌آی‌پی» نیز بیان‌کننده‌ی نوع میکروکنترلر آن است که اکنون به شرح آن پرداخته می‌شود.



شکل ۲.۱: آردوینو «اونو، آر۳، دی‌آی‌پی» (راست) و آردوینو «اونو، آر۳، اس‌ام‌دی» (چپ). توضیحات این بخش از کتاب، منطبق با بورد سمت راست ارایه می‌شوند.

عبارت «دی‌آی‌پی»[۴] (و به‌صورت عام «تی‌اچ‌تی»[۵]) معمولاً در مقابل عبارت

۱. Arduino UNO R3 DIP
۲. Uno
۳. Revision 3
۴. DIP (dual in-line package)
۵. THT (through-hole technology)



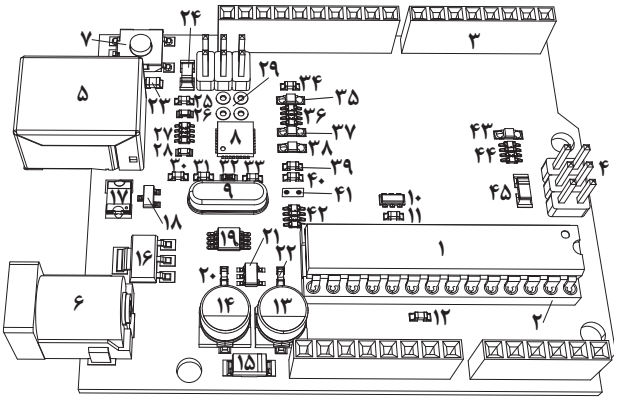
شکل ۲.۳: نحوه‌ی تعویض میکروکنترلر آردوینو

برای این منظور کافی است که میکروکنترلر آسیب‌دیده را به آرامی از سوکت آن بیرون کشید و قطعه‌ی سالم را به‌جای آن قرار داد. با توجه به وجود سوکت بر روی بورد، نیازی به لحیم‌کاری پایه‌ها نیست و فقط باید پایه‌ها را در جهت صحیح به داخل سوکت فشار داد. به غیر از این قطعه، سایر قطعات بورد آردوینو از مقاومت کافی در برابر اشتباهات متداول کاربران برخوردار بوده و ندرتاً آسیب می‌بینند. برای تعویض میکروکنترلر، بهتر است از تماس مستقیم دست با پایه‌ها اجتناب کرده و از دستکش و یا پنس پلاستیکی استفاده نمود تا الکتریسیته‌ی ساکن از بدن به قطعه منتقل نشود.

تقسیم وظایف و بخش‌بندی کلی سخت‌افزار

اگر بخواهیم سخت‌افزار آردوینو را به چند بخش اصلی تقسیم کنیم، می‌توانیم چهار بخش ذیل را به‌صورت کلّی برشماریم:

۱. میکروکنترلر
۲. بخش تغذیه
۳. بخش ارتباط با رایانه
۴. درگاه‌های گروهی

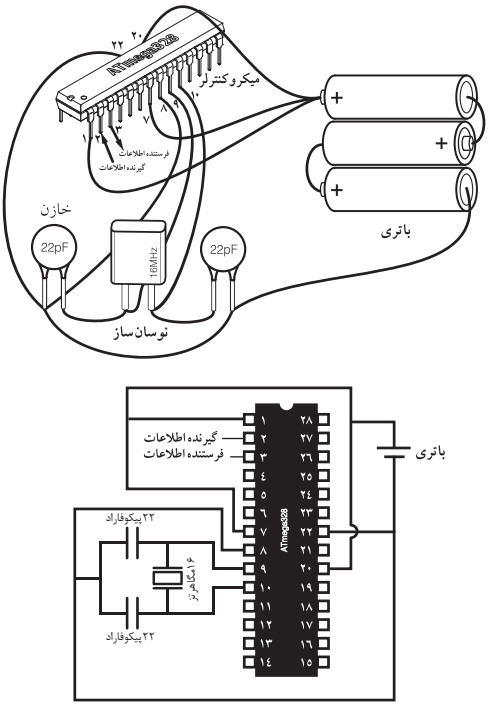


شکل ۲.۵: شماره‌گذاری قطعات الکترونیکی بورد آردوینو

جدول ۲.۱: قطعات الکترونیکی بورد آردوینو

شماره	نام قطعه	مشخصات	توضیحات
۱	میکروکنترلر	ATmega328	پردازش و حافظه
۲	سوکت	۲۸پایه	سهولت تعویض میکروکنترلر
۳	درگاه‌های ارتباطی	پین‌هدرمادگی	ارتباط با پایه‌های میکروکنترلر
۴	درگاه‌های ارتباطی	پین‌هدر نری	ارتباط سریال (بدون یواس‌بی) با میکروکنترلر
۵	درگاه یواس‌بی	USB port type B	ارتباط یواس‌بی رایانه با میکروکنترلر
۶	درگاه تغذیه	۵/۵ میلی‌متر	اتصال فیش باتری یا آداپتور برای تغذیه‌ی بورد
۷	کلید فشاری	تک‌سوییچ	ریست بورد و راه‌اندازی مجدد برنامه
۸	میکرو پردازشگر	ATmega16U2	مبدّل ارتباط سریال به یواس‌بی
۹	نوسان‌سازکریستالی	۱۶ مگاهرتز	ایجاد سیگنال زمان برای میکرو پردازشگر یواس‌بی
۱۰	نوسان‌سازسرامیکی	۱۶ مگاهرتز	ایجاد سیگنال زمان برای میکروکنترلر آردوینو (یک خازن ۱۰ پیکوفاراد نیز داخل قطعه تعبیه شده است)
۱۱	مقاومت	۱ مگااهم	حفظ تعادل نوسان در نوسان‌ساز سرامیکی
۱۲	خازن	۰/۱ میکروفاراد	حذف نویز از جریان تغذیه‌ی میکروکنترلر

زمان‌بندی و ایجاد پالس‌ساعت[1] در میکروکنترلر، پایه‌های ۹ و ۱۰ به نوسان‌ساز خارجی متصل می‌شوند. تکرار این نکته حایزاهمیت است که نصب میکروکنترلر بر روی بورد الکترونیکی، تنها به منظور سهولت استفاده از آن است.
میکروکنترلر آردوینو می‌تواند به‌تنهایی و فقط با بهره‌گیری از یک نوسان‌ساز خارجی (شامل یک نوسان‌ساز کریستالی و دو خازن‌سرامیکی) شروع به کار نماید (شکل ۲.۱۱).

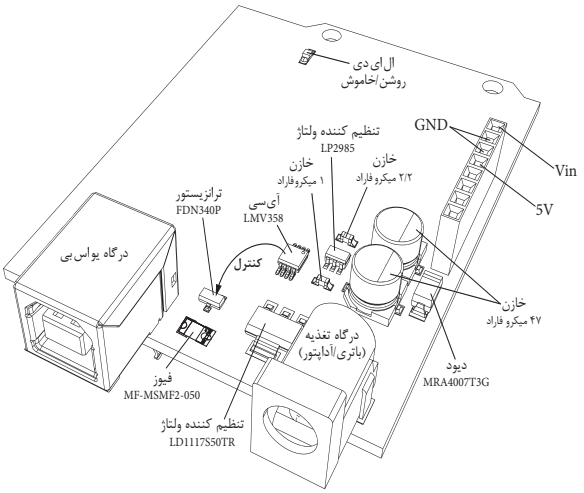


شکل ۲.۱۱: میکروکنترلر آردوینو می‌تواند با کمک یک نوسان‌ساز خارجی (یک کریستال و دو خازن) شروع به کار نماید.

برای این منظور ابتدا باید مطابق با شکل فوق، پایه‌های شماره ۹ و ۱۰ را به یک نوسان‌ساز کریستالی ۱۶ مگاهرتز متصل نمود و سپس هریک از پایه‌های نوسان‌ساز را نیز به یک خازن سرامیکی ۲۲ پیکوفاراد متصل کرد. خازن‌ها باید از محل اتصال به یکدیگر، به پایه‌های ۸ و ۲۲ (اتصال زمین) میکروکنترلر وصل

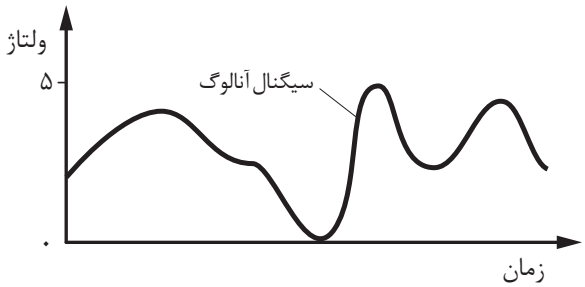
۱. Clock signal

- درگاه تغذیه (باتری/آداپتور)
- درگاه یواس‌بی
- درگاه Vin
- درگاه 5V

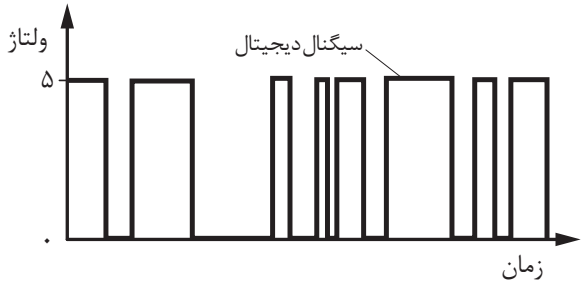


شکل ۲.۱۳: بخش تغذیه‌ی بورد آردوینو و اجزاء آن

درگاه تغذیه از طریق اتصال یک فیش ۵/۵ میلیمتری، به باتری و یا آداپتور ۶ تا ۲۰ ولت متصل شده و انرژی الکتریکی لازم برای راه‌اندازی آردوینو را تامین می‌کند. در اولین مرحله، جریان ورودی به آردوینو، از دیود MRA4007T3G عبور داده می‌شود تا مدار در مقابل ولتاژ معکوس تغذیه محافظت شود، به این معنا که اگر مثلاً قطب‌های مثبت و منفی باتری به‌صورت جابه‌جا به درگاه تغذیه وصل شده باشند، این دیود اجازه‌ی عبور جریان از باتری به سخت‌افزار را نمی‌دهد. توجه نمایید که سوراخ داخلی فیش ۵/۵ میلیمتری، قطب مثبت و بدنه‌ی خارجی آن قطب منفی است. سوالی که در این‌جا مطرح می‌شود آن است که: اگر ولتاژ مورد نیاز آردوینو ۵ ولت است پس چطور با ولتاژ ۲۰ ولت هم کار می‌کند؟ در اینجا وظیفه‌ی تنظیم‌کننده‌ی‌ولتاژ LD1117S50TR آشکار می‌شود. تنظیم‌کننده‌ی‌ولتاژ، ضمن کاهش ولتاژ ورودی به ۵ ولت، آن را ثابت و پایدار می‌سازد. دو خازن الکترولیت ۴۷ میکروفاراد نیز به فیلترکردن جریان



شکل ۲.۱۵: یک نمونه سیگنال آنالوگ (محور عمودی معرّف ولتاژ و محور افقی معرّف زمان است).

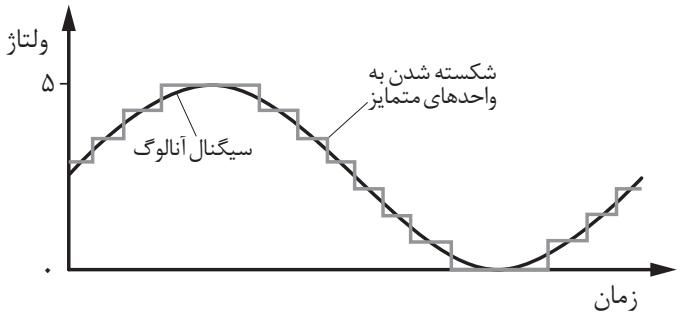


شکل ۲.۱۶: یک نمونه سیگنال دیجیتال (محور عمودی معرّف ولتاژ و محور افقی معرّف زمان است).

گروهی از قطعات الکترونیکی بر پایه‌ی سیگنال‌های آنالوگ فعالیت می‌کنند، مثلاً میکروفن (ادراک طیف وسیع تغییرات صدا). گروهی دیگر از قطعات الکترونیکی بر پایه‌ی سیگنال‌های دیجیتال فعالیت می‌کنند، مثلاً کلید فشاری (یا وصل است یا قطع). آنچه که برای میکروکنترلر آردوینو قابل ادراک است سیگنال‌های دیجیتال می‌باشد. سوالی که در اینجا قابل طرح است آن است که چگونه می‌توان سیگنال‌های آنالوگ یک میکروفن را برای آردوینو تعریف نمود. برای این منظور مداری داخل میکروکنترلر آردوینو تعبیه شده است که ADC[1] یا «مبدّل آنالوگ به دیجیتال» نام دارد. درگاه‌هایی که بر روی برد آردوینو با حرف A نام‌گذاری شده اند (A0 تا A5) امکان ارتباط با این مبدّل را دارا بوده و می‌توانند داده‌های آنالوگ را دریافت کنند. مبدّل آنالوگ به دیجیتال آردوینو، یک مبدّل ۱۰ بیتی است،

۱. Analog to Digital Converter (ADC)

یعنی می‌تواند سیگنال آنالوگ ورودی را با دقت 2^{10} یا ۱۰۲۴ واحد، بخش‌بندی کند. به عبارت دیگر، داده‌های بی‌نهایت موجود در یک سیگنالِ هموار و پیوسته‌ی آنالوگ، به حداکثر ۱۰۲۴ واحد متمایز، شکسته می‌شوند تا پس از تبدیل به یک سیگنال دیجیتال، برای آردوینو قابل ادراک شوند. در واقع، میکروکنترلر آردوینو نمی‌تواند بی‌نهایت داده‌ی موجود در یک سیگنال نرم آنالوگ را ادراک کند، لذا آن را به ۱۰۲۴ واحد پلکانی و متمایز تقلیل می‌دهد (شکل ۲.۱۷). سیگنال دیجیتال نهایی از طریق پردازش این واحدهای پلکانی استخراج می‌شود.



شکل ۲.۱۷: شکسته شدن سیگنال نرم آنالوگ به واحدهای پلکانی و متمایز

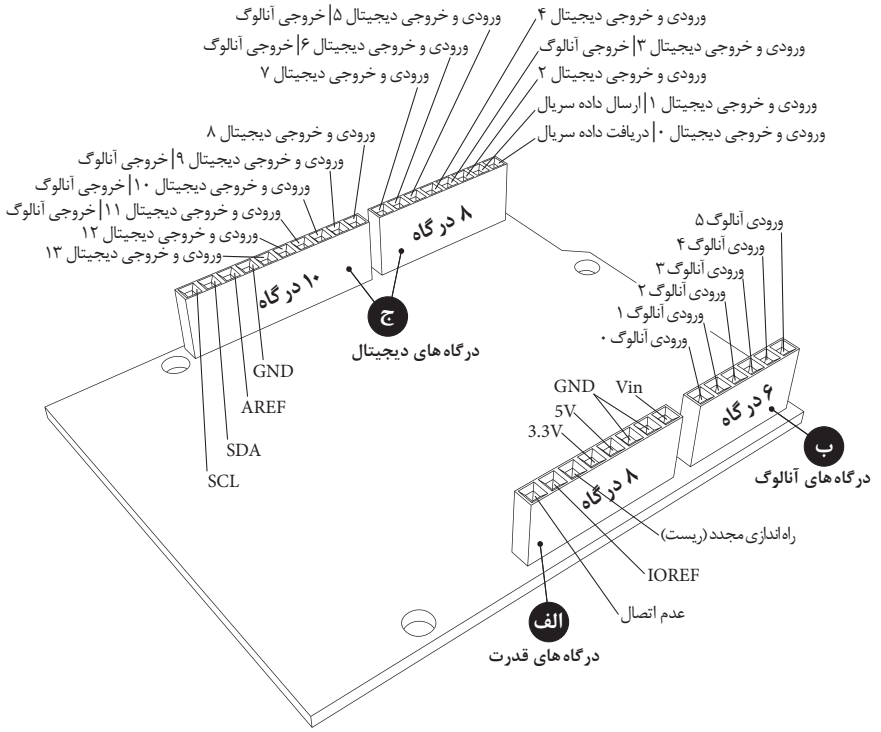
چالش دیگر آن است که آردوینو نمی‌تواند سیگنال‌های آنالوگ تولید نماید. مثلاً اگر بخواهیم یک ال‌ای‌دی فقط روشن و خاموش شود، به سیگنال‌های دیجیتال نیاز داریم که تولید این نوع سیگنال‌ها برای آردوینو بسیار آسان است. ولی اگر بخواهیم یک ال‌ای‌دی به آرامی کم‌نور شود، به سیگنال‌های آنالوگ نیاز داریم که آردوینو نمی‌تواند این نوع سیگنال را تولید کند. برای معادل‌سازی و تبدیل سیگنال‌های دیجیتال به آنالوگ، از تکنیکی به نام PWM[1] یا «مدولاسیون پهنای پالس» استفاده می‌کنیم. همان‌طور که پیش‌تر اشاره شد، سیگنال دیجیتال با پالس‌های منقطعِ خاموش یا روشن (مثلاً ۰ ولت یا ۵ ولت، و نه محدوده‌ی ولتاژ بین آن‌ها) ایجاد می‌شود. به «مدّت زمانی» که سیگنال در وضعیت روشن (۵ ولت) قرار دارد، اصطلاحاً «پهنای پالس» گفته می‌شود. برای معادل‌سازی سیگنال دیجیتال به آنالوگ، باید این پهنای پالس را تغییر دهیم که این تغییرات

۱. Pulse Width Modulation (PWM)

در اصطلاح، مدولاسیون نام دارند. اگر این تغییرات با سرعت بالایی بر پهنای پالس اعمال شوند، حالتی شبیه به سیگنال آنالوگ تداعی می‌گردد و مثلاً به نظر می‌رسد که ال‌ای‌دی به‌صورت پیوسته و نرم (و نه ناگهانی) کم‌نور می‌شود.

درگاه‌های گروهی

بر روی بورد آردوینو، چندین درگاه گروهی مادگی وجود دارند که از طریق آن‌ها می‌توان به تبادل جریان الکتریکی و نیز ارتباط دیجیتال و آنالوگ با سایر سخت‌افزارها، مثلاً حسگرها و عملگرها، پرداخت. برای این منظور کافی است که پین‌های نری یا سیم‌های تک‌رشته را به داخل درگاه مورد نظر فشار داد تا ارتباط الکتریکی ایجاد شود. این درگاه‌های گروهی با حروف الف تا ج بر روی شکل ۲.۱۸ نمایش داده شده اند.



شکل ۲.۱۸: درگاه‌های گروهی آردوینو

نبایـد هم‌زمـان از ۲۰۰ میلی‌آمپـر بیشـتر شـود.

گروه ب. درگاه‌های آنالوگ:

ایـن گـروه شـامل ۶ درگاه ورودی آنالـوگ اسـت کـه بـا اعـداد ۰ تـا ۵ شـماره‌گذاری شـده انـد (**A0**[1] **تـا A5**). مثـلاً اگـر بخواهیـم سـیگنال‌های آنالـوگ یـک حسـگر دمـا را بخوانیـم، بایـد آن را بـه یکـی از ایـن درگاه‌هـا متصـل نماییـم. بـرای تبدیـل سـیگنال‌های آنالـوگ بـه دیجیتـال (قابـل ادراک بـرای آردوینـو)، داده‌هـای وارد شـده بـه ایـن درگاه‌هـا توسـط ADC یـا مبـدّل آنالـوگ بـه دیجیتـال، معادل‌سـازی می‌شـوند کـه پیش‌تـر بـه شـرح آن پرداختـه شـد. لازم بـه ذکـر اسـت کـه درگاه‌هـای گـروه ب، در صـورت نیـاز، قابلیـت کاربـری به‌عنـوان درگاه‌هـای ورودی و خروجـی دیجیتـال را نیـز دارا می‌باشـند.

گروه ج. درگاه‌های دیجیتال:

ایـن گـروه شـامل ۱۸ درگاه اسـت کـه عمدتـاً درگاه‌هـای ورودی و خروجـی دیجیتـال می‌باشـند. مثـلاً اگـر بخواهیـم فرامیـن روشـن و خامـوش را از یـک کلیـد فشـاری دریافـت کنیـم، بایـد از درگاه‌هـای دیجیتـال اسـتفاده کنیـم. درگاه‌هـای ۲، ۴، ۷، ۸، ۱۲ و ۱۳ فقـط ورودی و خروجـی دیجیتـال هسـتند ولـی درگاه‌هـای ۰، ۱، ۳، ۵، ۶، ۹، ۱۰ و ۱۱ عـلاوه بـر ورودی و خروجـی دیجیتـال، وظایـف دیگـری را نیـز بـر عهـده دارنـد کـه به شـرح آن‌هـا پرداختـه می‌شـود.

0→RX[2]**:** ایـن درگاه کـه درگاه ۰ نـام دارد می‌توانـد عـلاوه بـر ورود و خـروج سـیگنال‌های دیجیتـال، بـه دریافـت داده‌هـای سـریال از سـایر سـخت‌افزارها نیز بپـردازد.

1→TX[3]**:** ایـن درگاه کـه درگاه ۱ نـام دارد می‌توانـد عـلاوه بـر ورود و خـروج سـیگنال‌های دیجیتـال، بـه ارسـال داده‌هـای سـریال بـه سـایر سـخت‌افزارها نیز بپـردازد.

2، 4، 7، 8، 12 و 13: ایـن درگاه‌هـا، ورودی و خروجـی دیجیتـال هسـتند یعنـی می‌تواننـد بـه دریافـت و ارسـال سـیگنال‌های دیجیتـال بپردازنـد. مثـلاً حسـگرها و عملگرهـای دیجیتـال بـه ایـن درگاه‌هـا متصـل می‌شـوند.

3~، 5~، 6~، 9~، 10~ و 11~: ایـن درگاه‌هـا عـلاوه بـر آن کـه ورودی و خروجـی دیجیتـال هسـتند، همچنیـن می‌تواننـد به‌عنـوان خروجـی آنالـوگ نیـز اسـتفاده شـوند. مثـلاً

۱. Analog (A)
۲. Receive (RX)
۳. Transmit (TX)

عملگرهـای آنالـوگ بـه ایـن درگاه‌هـا متصـل می‌شـوند. علامـت ~ در کنـار ایـن اعـداد، نشـانه‌ی آن اسـت کـه ایـن درگاه‌هـا از تکنیـک PWM یـا مدولاسـیون پهنـای پالـس، بـرای تبدیـل سـیگنال‌های دیجیتـال بـه آنالـوگ اسـتفاده می‌کننـد، کـه پیش‌تـر بـه شـرح آن پرداختـه شـد.

GND: مشخصات این درگاه منطبق با توضیحات نمونه‌ی مشابه در گروه الف است.

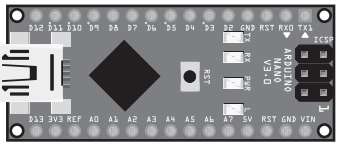
AREF[1]: ایـن درگاه، مرجـع «حـدّ ولتـاژ» بـرای مبـدّل آنالـوگ بـه دیجیتـال داخـل میکروکنترلـر اسـت. همان‌طـور کـه پیش‌تـر توضیـح داده شـد، ADC یـا مبـدّل آنالـوگ بـه دیجیتـال، بـرای تبدیـل پالس‌هـای آنالـوگ بـه پالس‌هـای قابـل درک بـرای آردوینـو (یعنـی پالس‌هـای دیجیتـال) بـه کار مـی‌رود. در بـورد آردوینـو (مـدل اونـو) به‌صـورت پیش‌فـرض، حـد بـالای دامنـه‌ی تغییـر ولتـاژ یـا «حـدّ ولتـاژ» در پالس‌هـای آنالـوگ، ۵ ولـت اسـت. حـال اگـر بخواهیـم کـه ADC دامنـه‌ی محدودتـری از ولتـاژ، مثـلاً ۰ تـا ۲ ولـت را در یـک سـیگنال آنالـوگ رصـد کنـد، بایـد ولتـاژی برابـر بـا ۲ ولـت را بـه درگاه AREF اعمـال کنیـم و بدیـن طریـق «حـدّ ولتـاژ» را بـه ۲ ولـت تغییـر دهیـم. مثـلاً اگـر یـک حسـگر دمـا داشـته باشـیم کـه سـیگنال‌های آنالـوگ خروجـی آن، بـر خـلاف معمـول در محـدوده‌ی ۵ ولـت نباشـد، بلکـه بـه ۲ ولـت محـدود شـود، بـرای آن کـه اندازه‌گیـری دمـا از دقـت بالاتـری برخـوردار باشـد، می‌توانیـم ۲ ولـت بـه درگاه AREF اعمـال کنیـم و بـه ایـن وسـیله بـه آردوینـو بفهمانیـم کـه دامنـه‌ی سـنجش سـیگنال‌ها را بـه ۰ تـا ۲ ولـت محـدود کنـد و روی ایـن محـدوده تمرکـز نمایـد. اگـر AREF آزاد باشـد و ولتـاژی بـه آن اعمـال نشـود، حـدّ ولتـاژ ۵ ولـت اسـت. دامنـه‌ی حـد ولتـاژ اعمـال شـده بـه ایـن درگاه بیـن ۰ تـا ۵ ولـت اسـت و نبایـد بـه بیشـتر از ۵ ولـت برسـد. فعال‌سـازی و کنتـرل ایـن درگاه از طریـق دسـتوری بـه نـام ()analogReference اسـت کـه در بخـش نرم‌افـزار آردوینـو بـه آن اشـاره می‌شـود.

SDA[2]: اگـر بخواهیـم تعـداد زیـادی حسـگر، عملگـر، و قطعـات الکترونیکـی را فقـط بـا اسـتفاده از دو رشـته سـیم بـه هـم وصـل کنیـم، یکـی از بهتریـن روش‌هـا، روشـی بـه نـام «یکپارچه‌سـازی مدارهـا»[3] یـا I2C اسـت. ایـن دو رشـته سـیم یـا دو خـط ارتباطـی عبارتنـد از: SDA (خـط تبـادل داده‌هـا) و SCL (خـط ارسـال پالـس سـاعت). بـر روی بـورد آردوینـو نیـز، دو درگاه بـه همیـن نام‌هـا وجـود دارنـد. مثـلاً اگـر بخواهیـم آردوینـو

۱. Analog Reference (AREF)
۲. Serial Data Line (SDA)
۳. Inter-Integrated Circuit (I2C or I²C)

آردوینو نانو[1]

نانـو مشـابه اونـو اسـت بـا ایـن تفـاوت کـه فشـرده‌تر و ارزان‌تـر اسـت. نانـو بـه آسـانی بـر روی بردبـورد سـوار می‌شـود. شـکل ظاهـری و مشـخصات عملکـردی آن بـه قـرار زیـر اسـت:



- نوع پردازشگر: ATmega168 و ATmega328P
- سرعت پردازشگر: ۱۶ مگاهرتز
- حجم حافظه:

-EEPROM: ۵۱۲/ کیلوبایت و ۱ کیلوبایت

-SRAM: ۱ کیلوبایت و ۲ کیلوبایت

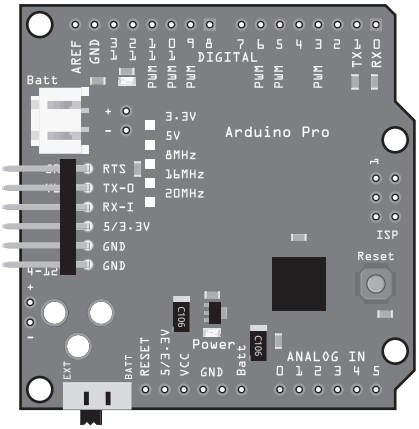
-فلش: ۱۶ کیلوبایت و ۳۲ کیلوبایت

- ولتاژ تغذیه: ۷ تا ۹ ولت
- تعداد درگاه ورودی آنالوگ: ۸
- تعداد درگاه خروجی آنالوگ: ۰
- تعداد درگاه ورودی و خروجی دیجیتال: ۱۴
- تعداد درگاه PWM: ۶
- درگاه مینی یواس‌بی
- ابعاد: ۴۵×۱۸ میلی‌متر

۱. Nano

آردوینو پرو[۱]

آردوینـو پـرو، سـخت‌افزاری فشـرده و کم‌ارتفاع اسـت و بـرای کاربران حرفه‌ای طراحی شـده اسـت. این مدل، گزینه‌ی مناسـبی برای اسـتفاده‌ی بلند مدت و دایمی در محصولات نهایی شـده و واقعی اسـت. ایـن نـوع بـورد، دارای درگاه‌هـای گروهـی مادّگـی نمی‌باشـد و خـود کاربـر باید درگاه‌هـا و یا سـیم‌های ارتباطـی را متناسـب بـا نیـاز خود در محل‌های مشـخص شـده لحیـم نماید. همچنین آردوینو پرو، فاقد درگاه یواس‌بی اسـت. شـکل ظاهری و مشـخصات عملکردی آن به قرار زیر اسـت:



- نوع پردازشگر: ATmega168 و ATmega328P
- سرعت پردازشگر: ۸ مگاهرتز و ۱۶ مگاهرتز
- حجم حافظه:

-EEPROM: ۵۱۲/ کیلوبایت و ۱ کیلوبایت

-SRAM: ۱ کیلوبایت و ۲ کیلوبایت

-فلش: ۱۶ کیلوبایت و ۳۲ کیلوبایت

- ولتاژ تغذیه: ۳/۳۵ تا ۱۲ ولت و ۵ تا ۱۲ ولت
- تعداد درگاه ورودی آنالوگ: ۶
- تعداد درگاه خروجی آنالوگ: ۰
- تعداد درگاه ورودی و خروجی دیجیتال: ۱۴
- تعداد درگاه PWM: ۶
- درگاه یواس‌بی: ندارد
- ابعاد: ۵۳×۵۲ میلی‌متر

۱. Pro

جدول تطبیقی بوردهای آردوینو

در جدول ۳.۱ بوردهای متداول آردوینو که در این فصل برشمرده شد، مورد مقایسه‌ی تطبیقی قرارگرفته‌اند.

جدول ۳.۱: جدول تطبیقی بوردهای آردوینو

مدل	پردازشگر		ولتاژ (ولت)	تعداد درگاه		حافظه (کیلو بایت)			درگاه پیاس‌بی
	نوع	سرعت (مگاهرتز)		آنالوگ ورودی/ خروجی	دیجیتال ورودی و خروجی/ PWM	EEPROM	SRAM	فلش	
اونو	ATmega328P	۱۶	۷–۱۲	۰/۶	۶/۱۴	۱	۲	۳۲	معمولی
نانو	ATmega168 ATmega328P	۱۶	۷–۹	۰/۸	۶/۱۴	۰/۵۱۲ ۱	۱ ۲	۱۶ ۳۲	مینی
مگا	ATmega2560	۱۶	۷–۱۲	۰/۱۶	۱۵/۵۴	۴	۸	۲۵۶	معمولی
پرو	ATmega168 ATmega328P	۸ ۱۶	۳/۳۵–۱۲ ۵–۱۲	۰/۶	۶/۱۴	۰/۵۱۲ ۱	۱ ۲	۱۶ ۳۲	ندارد
پرو مینی	ATmega328P	۸ ۱۶	۳/۳۵–۱۲ ۵–۱۲	۰/۶	۶/۱۴	۱	۲	۳۲	ندارد
لئوناردو	ATmega32u4	۱۶	۷–۱۲	۰/۱۲	۷/۲۰	۱	۲/۵	۳۲	میکرو
دوئه	AT91SAM3X8E	۸۴	۷–۱۲	۲/۱۲	۱۲/۵۴	۰	۹۶	۵۱۲	میکرو
میکرو	ATmega32U4	۱۶	۷–۱۲	۰/۱۲	۷/۲۰	۱	۲/۵	۳۲	میکرو
مینی	ATmega328P	۱۶	۷–۹	۰/۸	۶/۱۴	۱	۲	۳۲	ندارد
زیرو	ATSAMD21G18	۴۸	۷–۱۲	۱/۶	۱۰/۱۴	۰	۳۲	۲۵۶	میکرو
لیلی‌پَد	ATmega168V ATmega328P	۸	۲/۷–۵/۵	۰/۶	۶/۱۴	۰/۵۱۲	۱	۱۶	ندارد
یئون	ATmega32U4	۱۶	۵	۰/۱۲	۷/۲۰	۱	۲/۵	۳۲	میکرو
اِم‌کی‌آر–زیرو	SAMD21	۴۸	۳/۳	۱/۷	۱۲/۲۲	۰	۳۲	۲۵۶	میکرو
۱۰۱	Intel Curie	۳۲	۷–۱۲	۰/۶	۴/۱۴	۰	۲۴	۱۹۶	معمولی

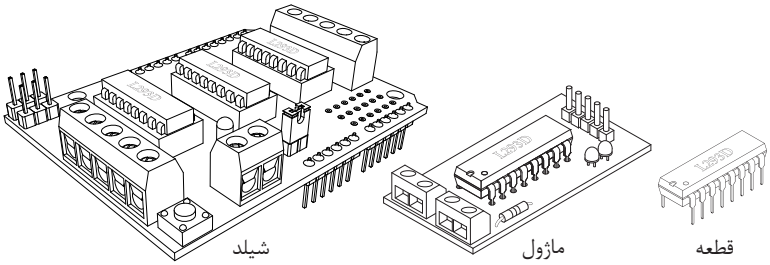
فصل ٤. ماژول‌ها و شیلدها

تعریف قطعه، ماژول و شیلد

عناصر سخت‌افزاری مرتبط با آردوینو را می‌توان به سه گروه اصلی تقسیم‌بندی کرد: قطعه[۱]، ماژول[۲]، و شیلد[۳]. قطعه عبارت از یک المان الکترونیکی یا مکانیکی تکی و ساده است، مثلاً مقاومت، آی‌سی، و استپرموتور. اگرچه هریک از این المان‌ها از چند جزء و یا قطعه‌ی دیگر تشکیل شده اند، لیکن تمام این اجزاء برای انجام عملکرد حداقلیِ قطعه‌ی الکترونیکی یا مکانیکی لازم بوده و عنصر اضافه‌ی دیگری، مثلاً برای سهولت کاربری و یا به‌عنوان ارزش افزوده، در کنار قطعه قرار نگرفته است. بنابراین یک عدد آی‌سی، به تنهایی، یک قطعه است. حال اگر همین آی‌سی، به‌عنوان کنترل‌کننده‌ی موتور، بر روی یک برد مدارچاپی لحیم شود و جهت حفاظت از آن در برابر جریان اضافی، یک قطعه مقاومت نیز به آن اضافه شده و برای ارتباط با کاربر، یک قطعه ال‌ای‌دی نشان‌گر برقراری جریان نیز در کنار آن قرارگیرد و نیز جهت سهولت اتصال سیم‌ها، چند درگاه نیز در کنار آن نصب شوند، اصطلاحاً به آن ماژول می‌گوییم؛ یعنی به‌جای «قطعه‌ی آی‌سی»، از عبارت «ماژول آی‌سی» استفاده می‌کنیم. حال اگر یک ماژول سخت‌افزاری، به گونه ای طراحی شود که بر روی بورد آردوینو سوار شده و اتصال آن از طریق پین‌های نری زیر آن با درگاه‌های گروهی آردوینو برقرار شود اصطلاحاً شیلد نامیده می‌شود؛ مثلاً «شیلد آی‌سیِ» راه‌انداز استپرموتور. در شکل ۴.۱ یک قطعه آی‌سی، یک ماژول آی‌سی، و یک شیلد آی‌سی برای راه‌اندازی و کنترل استپرموتور مشاهده می‌شوند.

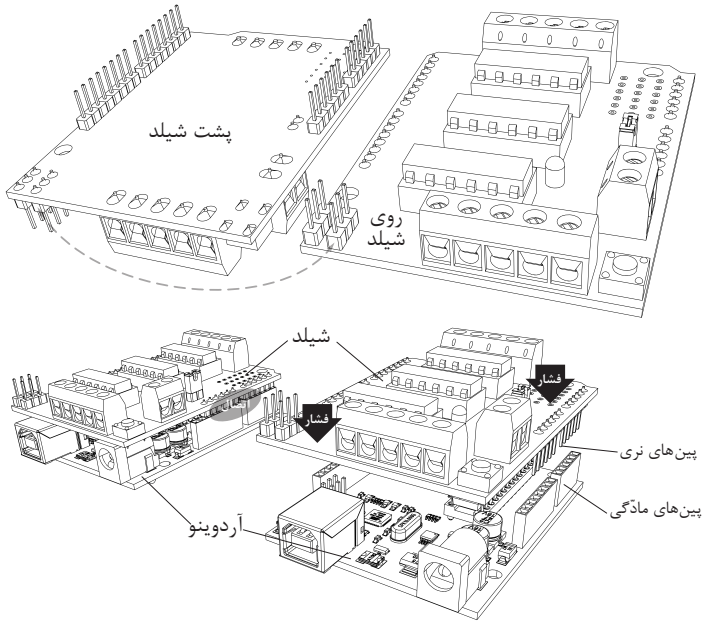
---

۱. Component
۲. Module
۳. Shield



شکل ۴.۱: یک قطعه آی‌سی، یک ماژول آی‌سی، و یک شیلد آی‌سی

نحـوه‌ی جای‌گـذاری و اتصـال شـیلد بـر روی آردوینـو نیـز در شـکل ۴.۲ نشـان داده شـده اسـت.



شکل ۴.۲: نحوه‌ی جای‌گذاری و اتصال شیلد بر روی بورد آردوینو

بـرای اتصـال شـیلد، بایـد آن را بـه گونـه‌ای بـر روی بـورد آردوینـو قـرار داد کـه پایه‌هـای نـری زیـر شـیلد، بـا اندکـی فشـار و بـدون خم‌شـدگی، بـا پین‌هـای مادّگـی (درگاه‌هـای گروهـی) درگیـر شـوند. برخـی از شـیلدها، عـلاوه بـر پین‌هـای نـری در زیـر بـورد،

حسگر دما

اگرچه حسگرهای مختلفی برای اندازه‌گیری دمای محیط وجود دارند ولی سه نمونه‌ی زیر، برای آردوینو متداول‌تر هستند:

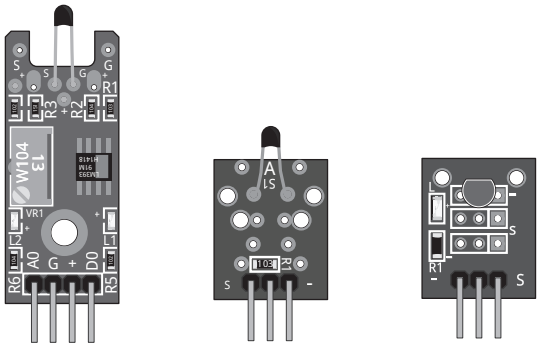
الف. ماژول KY-001 یک حسگر دیجیتال مجهز به DS18B20 با وضوح قابل برنامه‌ریزی ۹ تا ۱۲ بیت می‌باشد. محدوده‌ی سنجش دمای آن بین ۵۵- °C تا ۱۲۵ °C و دقّت اندازه‌گیری آن ۰/۵ ± °C است. ولتاژ عملکرد این ماژول بین ۳ تا ۵/۵ ولت بوده و دارای سه درگاه است: GND، 5V و سیگنال دیجیتال.

ب. ماژول KY-013 یک حسگر آنالوگ است و مجهز به ترمیستور[1] یا مقاومت الکتریکی حساس به دما است که با تغییرات دما، مقاومت الکتریکی آن تغییر می‌کند.. محدوده‌ی سنجش دمای آن بین ۵۵- °C تا ۱۲۵ °C و دقّت اندازه‌گیری آن ۰/۵ ± °C می‌باشد. ولتاژ عملکرد این ماژول ۵ ولت بوده و دارای سه درگاه است: GND، 5V و سیگنال آنالوگ.

ج. ماژول KY-028 یک حسگر دیجیتال است و مجهز به ترمیستور یا مقاومت الکتریکی حساس به دما است که با تغییرات دما، مقاومت الکتریکی آن تغییر می‌کند. این ماژول دارای هر دو نوع خروجی آنالوگ و دیجیتال است. بر روی این ماژول، پتانسیومتر کوچکی وجود دارد که با چرخاندن آن می‌توان آستانه‌ی تشخیص دما را برای خروجی دیجیتال تنظیم کرد. محدوده‌ی سنجش دمای آن بین ۵۵- °C تا ۱۲۵ °C و دقّت اندازه‌گیری آن ۰/۵ ± °C می‌باشد. ولتاژ عملکرد این ماژول بین ۳/۳ تا ۵/۵ ولت بوده و دارای چهار درگاه است: سیگنال آنالوگ، GND، 5V و سیگنال دیجیتال.

در شکل ۳.۴ تصاویر این سه ماژول آمده اند.

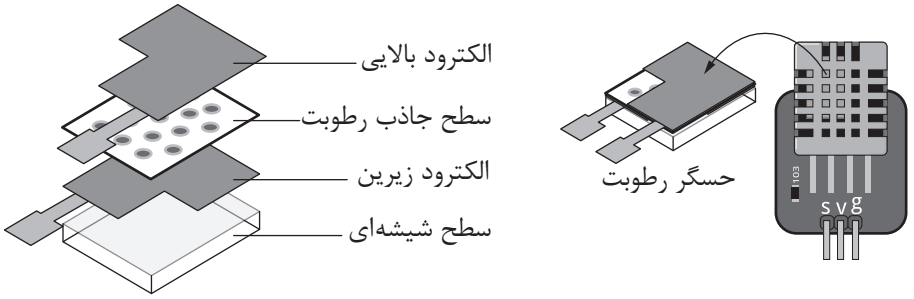
۱. Thermistor



شکل ۴.۳: سه نمونه‌ی مختلف از حسگرهای دما؛ الف. KY-001، ب. KY-013، و ج. KY-028

حسگر دما و رطوبت

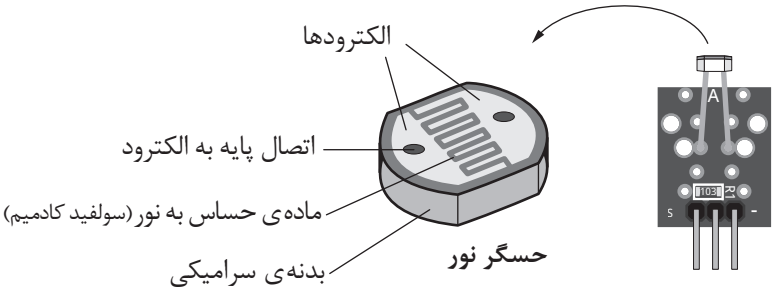
ماژول حسگر دما و رطوبت (مثلاً مدل‌های KY-015 یا DHT11 یا DHT22) امکان اندازه‌گیری هر دو پارامتر دما و رطوبت محیطی را دارا است. یک دماسنج مقاومتی و یک رطوبت‌سنج مقاومتی در داخل این ماژول قرارگرفته‌اند. رطوبت‌سنج مقاومتی از یک بستر شیشه‌ای و یک لایه‌ی جاذب رطوبت که در میان دو الکترود قرارگرفته‌اند تشکیل می‌شود. مقاومت الکتریکی لایه‌ی جاذب رطوبت، بسته به رطوبت دریافتی، تغییر می‌کند. آردوینو از طریق فرستادن و گرفتن سیگنال‌های دیجیتال، با این ماژول ارتباط برقرار می‌کند یعنی ابتدا سیگنال‌هایی را می‌فرستد و سپس منتظر پاسخ می‌ماند تا نهایتاً با جمع‌بندی داده‌ها به اطلاعات موردنظر برسد. محدوده‌ی سنجش دما در مدل DHT11 بین ۰ °C تا ۵۰ °C و دقّت اندازه‌گیری آن ۲ °C± است و محدوده‌ی سنجش رطوبت نسبی بین ۲۰% تا ۹۰% و دقّت اندازه‌گیری آن ۵% می‌باشد. ولتاژ عملکرد این ماژول بین ۳/۳ تا ۵/۵ ولت بوده و دارای سه درگاه است: GND، ۵V و سیگنال دیجیتال. در شکل۴.۴ تصویر این ماژول آمده است.



شکل ۴.۴: حسگر دما و رطوبت، و لایه‌های داخلی بخش سنجش رطوبت آن.

حسگر نور

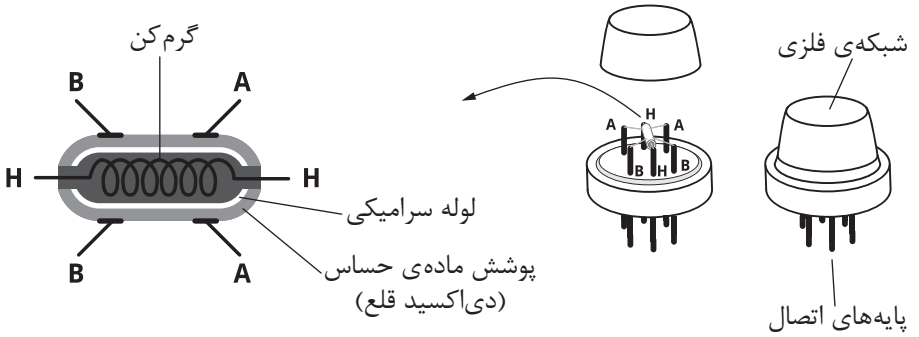
از ماژول حسگر نور (مثلاً مدل KY-018) برای سنجش شدت نور محیطی استفاده می‌شود. یک مقاومت الکتریکی حساس به نور[1] در داخل این ماژول قرارگرفته است. نحوه‌ی عملکرد حسگر به این صورت است که با افزایش نور محیط، مقاومت الکتریکی کاهش می‌یابد و با کاهش نور محیط، مقاومت الکتریکی افزایش می‌یابد. نتایج خروجی در قالب یک سیگنال آنالوگ به آردوینو فرستاده می‌شود. ولتاژ عملکرد این ماژول بین ۳/۳ تا ۵ ولت بوده و دارای سه درگاه است: GND، ۵V و سیگنال دیجیتال. در شکل ۴.۵ تصویر این ماژول آمده است.



شکل ۴.۵: حسگر نور و اجزاء متشکّله‌ی آن.

۱. LDR (Light Dependent Resistor) یا Photoresistor

برگرفته از حروف اوّل کلمات «Mingan Qilai» به زبان چینی است که معنی آن «حساس به گاز» می‌باشد). در شکل ۴,۱۲ تصویر یکی از نمونه‌های متداول این حسگرها به نام MQ-2 نشان داده شده است.



شکل ۴.۱۲: نمونه‌ای از حسگرهای تشخیص گاز، و ساختار داخلی آن.

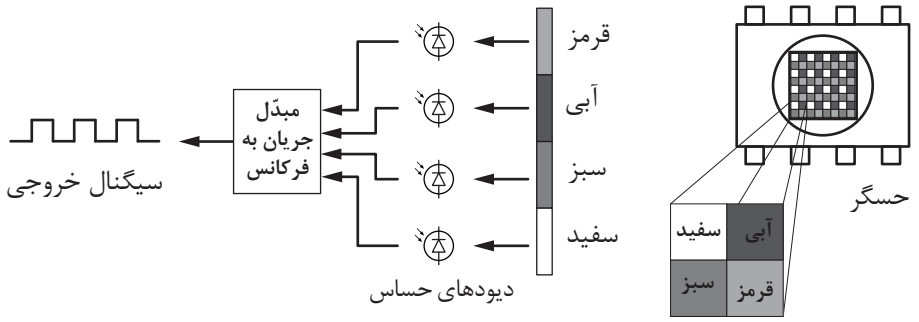
این حسگر از نوع آنالوگ بوده و مقاومت الکتریکی آن در مجاورت مواد شیمیایی موجود در محیط تغییر می‌کند. بخش حساس داخلی حسگر، از یک لوله سرامیکی تشکیل می‌شود. در داخل لوله، یک سیم‌پیچ قرار دارد که دو سر آن به پایه‌های H وصل شده اند و به‌عنوان گرم‌کن (هیتر) از آن استفاده می‌شود. سطح بیرونی لوله‌ی سرامیکی، با ماده‌ای حساس به نام دی‌اکسید قلع (SnO_2) پوشش یافته است. چهار سیم، با پوشش دی‌اکسید قلع در تماس‌اند که به پایه‌های A و B حسگر متصل شده اند. نحوه‌ی عملکرد حسگر به این صورت است که گرم‌کن داخلی، دی‌اکسید قلع را گرم می‌کند. اگر هوای اطراف، حاوی گازهای آلاینده باشد آن‌گاه دی‌اکسید قلع واکنش نشان داده و جریان الکتریکی برقرار می‌شود. معمولاً یک شبکه‌ی فلزی[۱] بر روی حسگر قرار می‌گیرد تا از ورود غبار به سطح حساس داخلی آن جلوگیری کرده و نیز مانع اشتعال گازهای محیطی توسط گرم‌کن داخلی حسگر شود. متداول‌ترین نمونه‌های حسگر تشخیص گاز و دود عبارتند از:

• حسگر MQ-2 برای تشخیص گازهای متان، بوتان، ال‌پی‌جی، و دود استفاده

۱. SUS316 mesh-100

حسگر تشخیص رنگ

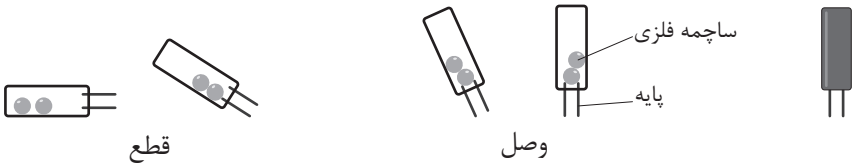
نورهـای رنگـی، ترکیبـی از سـه رنـگ اصلـی قرمـز، سـبز و آبـی هسـتند کـه با شـدت‌های مختلـف بـا هـم مخلـوط شـده‌اند. بنابرایـن می‌تـوان رنگ‌هـای پیچیـده را بـا اندازه‌گیـری مقـدار ایـن سـه رنـگ اصلـی موجـود در آن‌هـا شناسـایی نمـود. عملکـرد ماژول تشـخیص رنـگ TCS230 بـر همیـن اسـاس اسـتوار اسـت و می‌تـوان رنگ‌هـای مختلـف را بـر اسـاس طـول مـوج آن‌هـا شناسـایی کـرده و تشـخیص داد. ایـن مـاژول دارای چهـار ال‌ای‌دی بـا نـور سـفید اسـت کـه بـرای نورپـردازی شـیء مقابـل آن اسـتفاده می‌شـود. همچنیـن یـک تراشـه‌ی حسـاس بـه رنگ‌هـای اصلـی نـور، بـر روی ایـن مـاژول تعبیـه شـده اسـت. تراشـه‌ی مذکـور شـامل یـک ماتریـس ۸×۸ (۶۴ عـدد) از دیودهـای حسـاس بـه نورهـای قرمـز، سـبز، آبـی و سـفید اسـت. یعنـی بـرای خوانـدن هریـک از ایـن رنگ‌هـای اصلـی، ۱۶ دیـود اختصاصـی وجـود دارد. داده‌هـای خوانـده شـده توسـط ایـن دیودهـا، بـا کمـک یـک مبـدّل جریـان بـه فرکانـس، بـه سـیگنالی تبدیـل می‌شـود کـه فرکانـس آن متناسـب بـا شـدّت نـور آن رنـگ مـورد نظـر اسـت. ایـن فرکانـس توسـط آردوینـو خوانـده خواهـد شـد (شـکل ۴.۱۸).



شکل ۴.۱۸: حسگر رنگ شامل یک ماتریس ۸×۸ از دیودهای حساس به نورهای قرمز، آبی، سبز و سفید است. داده‌های خوانده شده از دیودها، با کمک مبدّل جریان به فرکانس، به سیگنالی تبدیل می‌شود که فرکانس آن سیگنال متناسب با شدّت نور رنگ مورد نظر است.

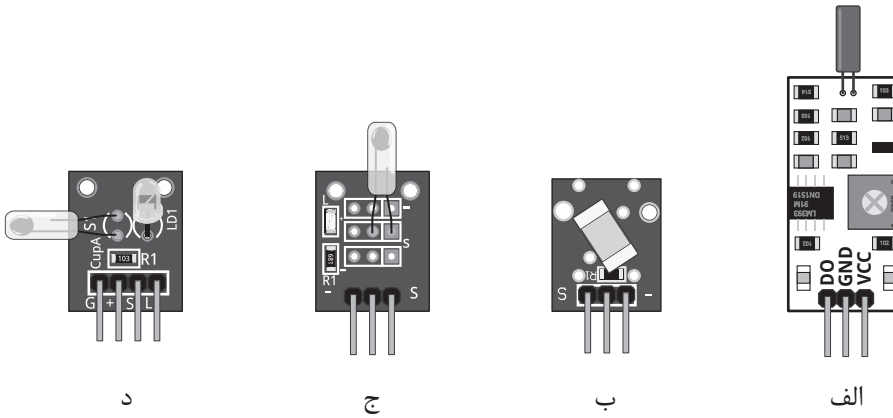
ولتاژ عملکرد این حسگر بین ۲/۷ تا ۵/۵ ولت بوده و دارای هشت درگاه است:

- درگاه GND
- درگاه فعال‌سازی خروجی



شکل ۴.۲۰: نحوه‌ی عملکرد حسگر شیب ساچمه‌ای: ساچمه‌ها می‌لغزند و در پایین لوله، روی میله‌های رسانا قرار می‌گیرند.

ماژول مشابه دیگری به نام KY-017 وجود دارد که به‌جای ساچمه‌ی فلزی، از یک قطره فلز جیوه بهره می‌گیرد. ولتاژ عملکرد این ماژول‌ها بین ۳/۳ تا ۵ ولت بوده و دارای سه درگاه می‌باشند: GND، ۵V و سیگنال دیجیتال خروجی. ماژول دیگری به نام KY-027 بر همین اساس ساخته شده است که با حرکت دادن آن، یک ال‌ای‌دی روشن و خاموش می‌شود. در شکل ۴.۲۱ تصاویر این چهار ماژول آمده اند.



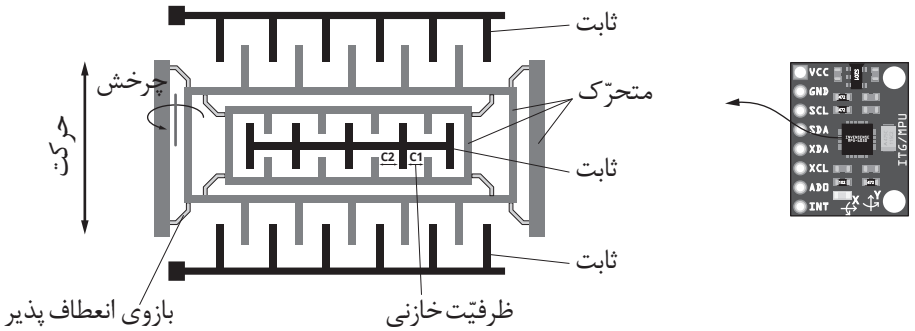
الف ب ج د

شکل ۴,۲۱: حسگرهای مختلف شیب؛ الف. SW-520D، ب. KY-020، ج. KY-017، و د. KY-027.

حسگر لرزش

ماژول حسگر لرزش (مثلاً مدل KY-002 یا KY-031) دارای یک فنر رسانای الکتریکی است که با تکان خوردن آن، اتصال الکتریکی برقرار می‌شود. حساسیت این فنر نسبت به لرزش، ضربه و شوک‌های حرکتی، مبنای عملکرد این حسگر است. ولتاژ عملکرد این ماژول ۵ ولت بوده و دارای سه درگاه می‌باشد: GND،

یکی از متداول‌ترین آن‌ها در شکل ۴.۲۷ آمده است. در مرکز حسگر، جرم کوچکی وجود دارد که می‌تواند به آسانی حرکت کند و مرتعش شود. در اطراف این جرم، ساختار انعطاف پذیری وجود دارد و زمانی‌که سرعت زاویه‌ای از محیط خارج به حسگر وارد شود، این ساختار انعطاف‌پذیر، در اطراف جرم شروع به حرکت می‌کند. حرکت این ساختار انعطاف‌پذیر نسبت به جرم، ظرفیت خازنی بین آن‌ها را تغییر می‌دهد و می‌توان بر اساس این تغییرات، سرعت زاویه‌ای را در یک تا سه محور مختلف[1] اندازه‌گیری کرد و نیز عدم تعادل جسم را تشخیص داد.



شکل ۴.۲۷: ژیروسکوپ و نحوه‌ی عملکرد آن: سرعت زاویه‌ای سبب تغییر ظرفیّت خازنی بین تیغه‌های ثابت و متحرک می‌شود.

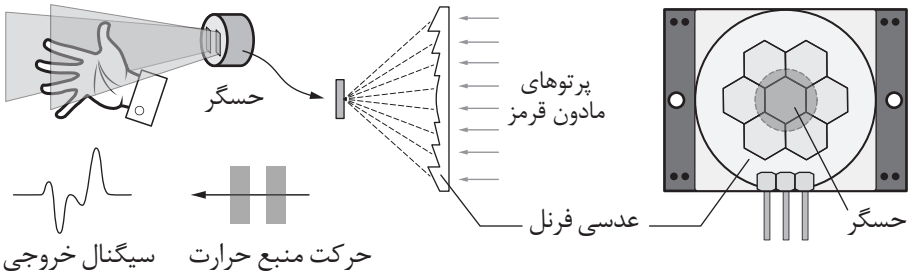
ولتاژ عملکرد ماژول ژیروسکوپ بین ۲/۷ تا ۶/۵ ولت بوده و دارای هشت درگاه است: GND، درگاه تغذیه، درگاه انتخاب نوع تبادل داده‌ها، پالس ساعت، ورودی و خروجی داده‌های سریال، خروجی داده‌های سریال، و درگاه‌های ارسال وقفه ۱ و ۲. در شکل ۴.۲۷ تصویر این ماژول آمده است.

کلید فشاری

ماژول کلید فشاری (مثلاً مدل KY-004) مشتمل بر یک کلید فشاری است که به‌عنوان یک حسگر دیجیتال ساده عمل می‌کند و دو نوع خروجی می‌تواند داشته باشد: LOW و HIGH. زمانی‌که کلید آزاد است مقدار HIGH از آن خوانده

۱. Yaw, Pitch, and Roll

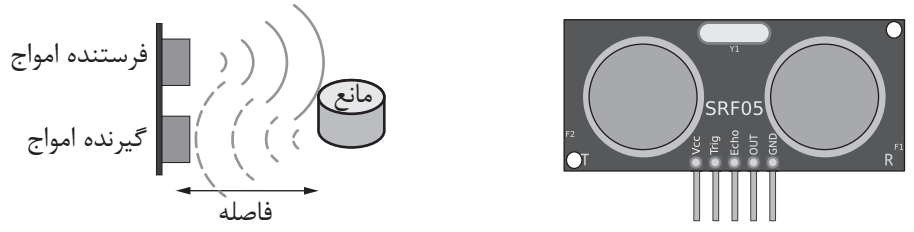
می‌شود. محدوده‌ی شناسایی حرکت، محدود به ۱۰ متر و حدود ۳۰۰ درجه است. محدوده‌ی ولتاژ عملکرد این ماژول بین ۵ تا ۱۲ ولت است و دارای سه درگاه می‌باشد: تغذیه، GND، و خروجی دیجیتال. در شکل ۴.۳۹ تصویر این ماژول آمده است.



شکل ۴.۳۹: حسگر تشخیص حرکت و نحوه‌ی عملکرد آن.

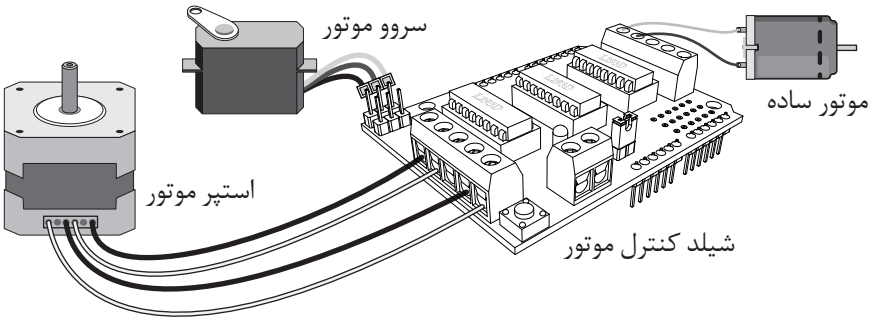
حسگر فاصله‌یاب فراصوتی

ماژول اولتراسونیک (مثلاً مدل KY-050 یا HC-SR04) دارای یک فرستنده و یک گیرنده‌ی امواج مافوق صوت است. امواج ابتدا به سمت مانع فرستاده می‌شوند و پس از برخورد به آن باز می‌گردند. بر مبنای زمان پژواک موج، می‌توان فاصله‌ی حسگر تا مانع را محاسبه کرد. این نوع حسگرها تحت تاثیر رنگ و نور اشیاء مختلف گمراه نمی‌شوند ولی در برابر اجسام متخلخل، از کارایی بالایی برخوردار نیستند. محدوده‌ی شناسایی حسگر بین ۲ سانتی‌متر تا ۴ متر و حدود ۳۰ درجه است. ولتاژ عملکرد این ماژول ۵ ولت است و دارای چهار درگاه می‌باشد: تغذیه، ورودی دیجیتال، خروجی دیجیتال، و GND. در شکل ۴.۴۰ تصویر این ماژول آمده است.



شکل ۴.۴۰: حسگر فاصله‌یاب فراصوتی و نحوه‌ی عملکرد آن.

برای کنترل هر سه نوع موتور فوق توسط آردوینو، می‌توان از شیلد کنترل موتور استفاده کرد. این شیلد از مدارهای مجتمعی مانند L293D و یا TB6612 بهره‌می‌گیرد که آی‌سی‌های راه‌انداز موتور هستند. توسط شیلد کنترل موتور می‌توان به‌آسانی تا ۴ موتور الکتریکی ساده ۴/۵ تا ۱۳/۵ ولت یا موتور سروو و یا ۲ استپر موتور را راه‌اندازی نمود. قابلیت مدیریت چندین موتور توسط این شیلد آن‌را برای ساخت پروژه‌های پیچیده مناسب می‌سازد. در شکل ۴.۴۷ تصویر این شیلد آمده است.



شکل ۴.۴۷: شیلد کنترل موتور

ماژول رله

برای کنترل ولتاژهای بالا توسط آردوینو، از ماژول رله (مثلاً مدل KY-019) استفاده می‌کنیم. مثلاً اگر بخواهیم روشن و خاموش‌شدن چراغ‌های خانه را که با ولتاژ ۲۲۰ ولت کار می‌کنند، توسط آردوینو کنترل‌کنیم باید از ماژول رله، به‌عنوان واسط بین آردوینو و چراغ‌های برق، استفاده کنیم. رله‌ها در دو نوع کلی الکترومکانیکی[۱] و الکترونیکی[۲] وجود دارند.

رله‌ی مکانیکی، دارای یک سیم‌پیچ است که با ولتاژ پایین (مثلاً ۵ ولت) ایجاد میدان مغناطیسی می‌کند و زبانه‌ای فلزی را قطع و وصل می‌کند که حامل ولتاژ بالا (مثلاً ۲۲۰ ولت) است. این نوع از رله‌ها دارای درگاه‌های زیر اند:

۱. Electromechanical Relay (EMR)
۲. Solid-State Relay (SSR)

فصل ۵. نرم‌افزار آردوینو

برای آن‌که به سخت‌افزار آردوینو بفهمانیم که چه کاری را باید انجام دهد، از برنامه‌نویسی رایانه‌ای بهره می‌گیریم. اصول برنامه‌نویسی در محیط آردوینو، مشابه برنامه‌نویسی به زبان C++ است و با استفاده از نرم‌افزار اختصاصی آردوینو که نام آن Arduino IDE است انجام می‌شود. نرم‌افزار Arduino IDE برای کدنویسی و نیز ترجمه‌ی کدها به زبان قابل‌درک برای سخت‌افزار آردوینو طراحی شده‌است. عبارت IDE مخفف کلمات Integrated Development Environment است که به‌معنای «محیط توسعه‌ی یکپارچه» می‌باشد. نرم‌افزار Arduino IDE رایگان و متن‌باز بوده و بر روی سیستم‌عامل‌های مختلف (ویندوز، مکینتاش و لینوکس) قابل نصب است.

عملیاتی که توسط نرم‌افزار آردوینو انجام می‌شود به این صورت است که مثلاً می‌خواهیم روشن و خاموش‌شدن یک ال‌ای‌دی را کنترل نماییم. ابتدا باید دستورات لازم برای این کار را در محیط نرم‌افزار بنویسیم. سپس نرم‌افزار باید این دستورات را به کدهای قابل‌درک برای سخت‌افزار تبدیل کند. این کدها توسط سخت‌افزار دریافت شده و در حافظه‌ی آن ذخیره می‌شوند. سپس پردازشگر سخت‌افزار، مطابق با برنامه‌ی نوشته شده، دستورات لازم را به درگاه‌های خروجی می‌فرستد و ال‌ای‌دی را کنترل می‌کند.

نصب نرم‌افزار آردوینو

ابتدا باید نرم‌افزار آردوینو را بر روی رایانه‌ی خود نصب کنید. برای دریافت نرم‌افزار، ابتدا به آدرس زیر مراجعه نمایید:

https://www.arduino.cc/en/Main/Software

سپس گزینه‌ی متناسب با سیستم‌عامل کامپیوتر یا لپ‌تاپ خود را انتخاب کنید. مثلاً اگر سیستم‌عامل رایانه‌ی شما ویندوز است، گزینه‌ی مربوط به ویندوز را

انتخـاب نماییـد و فایـل را دانلـود کنیـد. فایـل دانلـود شـده را بـر روی رایانـه‌ی خـود نصـب نماییـد. لازم نیسـت کـه تنظیمـات پیش‌فـرض نصـب را تغییـر دهیـد.
سـخت‌افزار آردوینـو را بـا اسـتفاده از سـیم رابـط بـه یکـی از درگاه‌هـای یواس‌بـی رایانـه‌ی خـود وصـل کنیـد. چـراغ ON بـر روی بـورد آردوینـو روشـن می‌شـود. در ایـن زمـان، سیسـتم‌عامل رایانـه‌ی شـما به‌صـورت خـودکار اقـدام بـه شناسـایی بـورد و نصـب درایورهـای مربوطـه می‌کنـد. جهـت اطمینـان از شناسـایی بـورد آردوینـو توسـط رایانـه، بهتـر اسـت مراحـل زیـر را نیـز انجـام دهیـد:
مثـلاً اگـر سیسـتم‌عامل رایانـه‌ی شـما ویندوز اسـت ابتـدا بـه قسـمت Start Menu ویندوز برویـد. سـپس Control Panel را بـاز کنیـد و بـه بخـش System and Security برویـد و گزینه‌ی System را انتخـاب کنیـد. سـپس Device Manager را بـاز کنیـد. اکنـون بایـد در یکـی از زیـر شـاخه‌های Ports (COM & LPT) یـا Other Devices گزینـه‌ی جدیـدی را ببینیـد کـه بـا یکی از ایـن دو نـام مشـخص شـده اسـت: Arduino UNO (COMxx) یـا Unknown Device. بر روی ایـن گزینـه‌ی جدیـد، کلیک‌راسـت کنیـد و دسـتور Update Driver Software را انتخـاب کنید. در پنجره‌ی جدید، Browse my computer for Driver software را انتخاب نموده و سپس وارد پوشـه‌ای شـوید کـه نرم‌افـزار آردوینـو را در آن نصـب کرده‌ایـد. از داخـل پوشـه‌ی Drivers فایـل ArduinoUNO.inf را انتخـاب کنیـد تـا ویندوز به‌طـور خـودکار، کار نصـب و شناسـایی سـخت‌افزار آردوینـو را انجـام دهـد. اگـر فایـل مذکـور را نمی‌یابیـد، احتمـال دارد کـه فایـل بـه حالـت مخفـی در آمـده باشـد کـه بایـد آن را از ایـن حالت خـارج کنید.
در ایـن قسـمت، بـرای اطمینـان از صحـت عملکـرد و ارتبـاط سـخت‌افزار و نرم‌افـزار آردوینـو، به‌صـورت زیـر عمـل می‌نماییـم:
نرم‌افـزار را اجـرا کنیـد و وارد آن شـوید. از طریـق منـوی بـالای نرم‌افـزار و مطابـق بـا مسـیر زیـر، فایـل Blink را انتخـاب کنیـد:

File > Examples > 1.Basics > Blink

سـپس از طریـق مسـیر زیـر، نـوع بـورد آردوینـوی خـود را بـه نرم‌افـزار معرفـی نماییـد (شـما بایـد از داخـل لیسـت، بـورد Arduino Uno را انتخـاب کنید):

Tools > Board > your board type

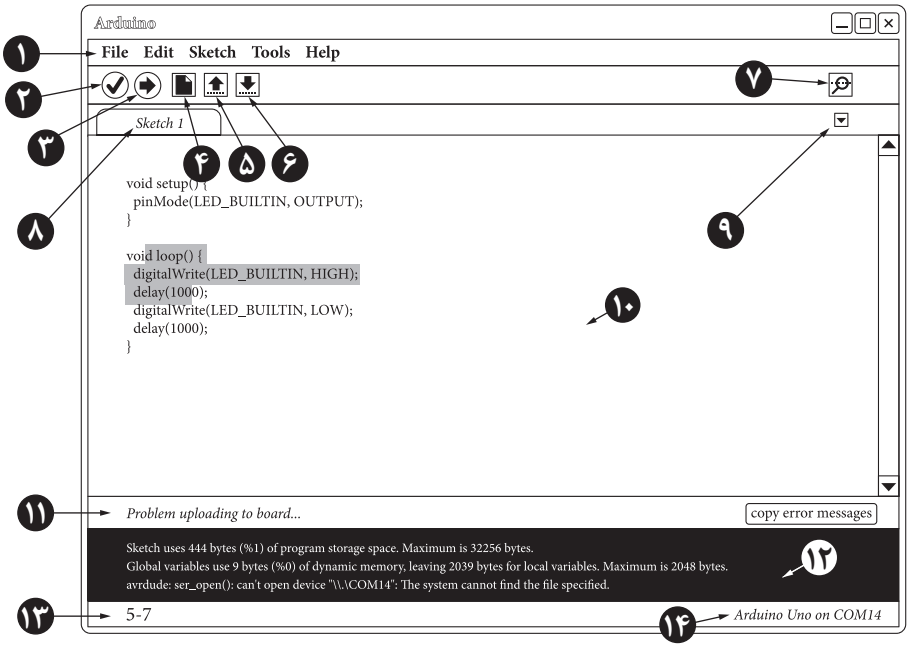
اکنون از طریق مسیر زیر، به نرم‌افزار بگویید که بورد آردوینو به کدام درگاه کامپیوتر متصل است:

Tools > Port > COMxx

اگـر نمی‌دانیـد که در لیسـت موجود بایـد کدام درگاه را انتخاب کنید، سـیم رابـط را از بورد آردوینـو جـدا کنیـد و مجدداً وصل کنید. درگاهـی که نام آن برای لحظه‌ای از لیسـت حذف می‌شـود و مجـدداً ظاهـر می‌شـود، همـان درگاه مـورد نظـر اسـت کـه بایـد انتخـاب شـود. اکنـون درحالی‌کـه سـخت‌افزار آردوینـو بـه کامپیوتـر متصـل اسـت و نیـز فایـل Blink بـر روی نرم‌افـزار آردوینـو بـاز اسـت، دکمـه‌ی Upload (شـکل آن شـبیه فلش‌افقـی اسـت) را در داخـل نرم‌افـزار فشـار دهیـد. پـس از چنـد لحظـه بایـد چـراغ کوچکـی کـه بـر روی بـورد آردوینـو وجـود دارد شـروع بـه چشـمک‌زدن نمایـد. درصورتی‌کـه چـراغ چشـمک نمی‌زنـد، مراحـل بـالا را مجـدداً کنتـرل نمـوده، و نیـز بـا اندکـی فشـار بـه اتصـالات کابـل، از اتصـال کامـل آن بـا درگاه‌هـای یواس‌بـی مطمئـن شـوید.

آشنایی با محیط نرم‌افزار آردوینو

محیط نرم‌افزار آردوینو (شکل ۵.۱) از چند بخش اصلی تشکیل شده است که در این قسمت به شرح این بخش‌ها پرداخته می‌شود.



شکل ۵.۱: محیط نرم‌افزار آردوینو و بخش‌های اصلی آن

۱. نوار فهرست[۱]: در این بخش، دسترسی به امکاناتی فراهم می‌شود که برای ایجاد و ذخیره‌ی اسکچ‌های آردوینو لازم اند. همچنین نمونه‌کدها و مثال‌ها، کتابخانه‌ها، درگاه‌های رایانه، فهرست بوردهای آردوینو، راهنمای نرم‌افزار، و غیره در این قسمت قرار دارند.

۲. دکمه‌ی تأیید: فشردن این دکمه باعث می‌شود که کدهای نوشته شده توسط شما، کامپایل شده و چنانچه اشتباهات نوشتاری یا دستوری در متن آن وجود داشت، مشخص شوند.

۳. دکمه‌ی بارگذاری: فشردن این دکمه، کدهای نوشته شده توسط شما را بر روی بورد سخت‌افزاری آردوینو بارگذاری می‌کند.

۱. Menu bar

جدول ۵.۱: برخی از علایم و نشانه‌های مورد استفاده در برنامه‌نویسی آردوینو

علامت	نام فارسی	نام انگلیسی
,	ویرگول	Comma
;	نقطه ویرگول	Semicolon
*	ستاره	Asterisk
" "	علامت گفتاورد	Quotation mark
/	خط مورب	Slash
\	واکج‌خط	Backslash
_	زیرخط	Underscore
-	خط‌فاصله	Dash
( )	پرانتز	Round brackets
[ ]	کروشه	Square brackets
{ }	آکولاد	Curly brackets
< >	پرانتز شکسته	Angle brackets
<	کوچک‌تر	Less-than
>	بزرگ‌تر	Greater-than

هریک از این علایم و نشان‌ها، معنا و کاربرد خاص خود را دارند و اگر اشتباهاً به‌جای یکدیگر استفاده شوند، برنامه، قابل پردازش و استفاده نمی‌باشد. مثلاً همان‌طور که در پایان جملات نامه‌ها و متون روزمره نقطه می‌گذاریم، در پایان هر دستور آردوینو از علامت (;) استفاده می‌شود. حال اگر در نوشتن دستورات، اشتباهاً (,) را به‌جای (;) تایپ کنید برنامه اجرا نمی‌شود و با پیغام خطا مواجه می‌شوید.

ثابت‌ها و متغیرها

در آردوینـو، مجموعـه‌ای از کلیـد واژه‌هـای از پیـش تعریف‌شـده وجـود دارنـد کـه هریـک از آن‌هـا معنـای ثابـت و مشـخصی دارنـد؛ مثـلاً:

کلید واژه‌ی HIGH برای روشن‌کردن درگاه

کلید واژه‌ی LOW برای خاموش‌کردن درگاه

کلید واژه‌ی INPUT برای ورودی‌کردن درگاه

کلید واژه‌ی OUTPUT برای خروجی‌کردن درگاه

کلیـد واژه‌ی LED_BUILTIN بـرای آدرس‌دهـی ال‌ای‌دی L (موجود بـر روی بورد آردوینو) بـه ایـن کلیـد واژه‌هـا کـه معنـا و مفهـوم آن‌هـا همیشـه ثابـت اسـت و تغییـر نمی‌کنـد، اصطلاحـاً «ثابـت» گفتـه می‌شـود.

نمونـه‌ی دیگـری از ثابت‌هـا، ثابت‌هـای عـددی هسـتند. ثابت‌هـای عـددی، اعـدادی هسـتند کـه به‌صـورت مسـتقیم در اسـکچ‌ها بـه کار می‌رونـد، مثـلاً عـدد ۲۷ یـک ثابـت عـددی اسـت و مقـدار آن در سراسـر اسـکچ تغییـر نمی‌کنـد. ثابت‌هـای عـددی می‌تواننـد از بیـن اعـداد صحیـح و اعـداد اعشـاری انتخـاب شـوند. در آردوینـو می‌تـوان اعـداد صحیـح را عـلاوه بـر مبنـای ۱۰ در مبناهـای ۲، ۸ و ۱۶ نیـز تعریـف کـرد. اعـداد اعشـاری را نیـز می‌تـوان در قالـب نمـاد علمـی نشـان داد.

«متغیر»هـا، برخـلاف ثابت‌هـا، مقادیـر مختلفـی را بـه خـود می‌گیرنـد و مقـدار آن‌هـا در طـی رونـد اجـرای اسـکچ تغییـر می‌کنـد. در واقـع، متغیرهـا فضاهایـی از حافظـه‌ی آردوینـو هسـتند کـه توسـط مـا نام‌گـذاری می‌شـوند تـا داده‌هـای اسـکچ مـا در آن‌هـا ذخیـره شـوند. ایـن داده‌هـا می‌تواننـد در خـلال اجـرای اسـکچ، خوانـده شـده و یـا تغییـر کننـد. نـام متغیـر بـه دلخـواه مـا انتخـاب می‌شـود: مثـلاً ABC یـا Example. زمانی‌کـه مـا متغیـری را تعریـف می‌کنیـم، بایـد «نـوع» آن را نیـز بـرای آردوینـو مشـخص نماییـم بـه ایـن معنـا کـه بگوییـم می‌خواهیـم چـه نـوع داده‌هایـی را بـه متغیـر اختصـاص دهیـم و در آن ذخیـره نماییـم. نمونه‌هایـی از «نـوع داده‌هـا» عبارتنـد از:

char: مخفـف Character اسـت و در زبـان فارسـی، نویسـه نامیـده می‌شـود و مشـتمل بـر حـرف، رقـم و نشـانه‌ها اسـت. ایـن نـوع می‌توانـد یـک نویسـه‌ی تک‌حرفـی (مثـلاً A) و یـا یـک نویسـه‌ی یـک تـا سـه رقمـی بیـن ۱۲۸- تـا ۱۲۷ (مثـلاً ۶۵) را در خـود

ذخیره کند و یا اصطلاحاً به خود «انتساب» دهد. توجه داشته باشید که در زمان انتساب مقدار، لازم است:
نویسه‌های تک‌حرفی در داخل علامت (' ') نوشته شوند. مثلاً 'A'
نویسه‌های رقمی، بدون علامت نوشته شوند. مثلاً ۶۵

```
char Example = 'A'
char Example = 65
```

انتساب مقدار یعنی مقادیر 'A' و ۶۵ به متغیر Example اختصاص داده شده و در آن ذخیره گردیده‌اند. درباره‌ی مفهوم انتساب در بخش بعد صحبت می‌شود.

int: مخفف Integer به‌معنای عددصحیح است و متداول‌ترین نوع داده در نگارش اسکچ‌های آردوینو محسوب می‌شود. در این نوع داده، می‌توان یک عددصحیح بین ۳۲۷۶۸- تا ۳۲۷۶۷ را ذخیره نمود. اگر نمی‌دانید چه نوع داده‌ای برای متغیر مناسب است، از نوع int استفاده کنید. مثال:

```
int Example = -9
```

unsigned int: این نوع داده، مشابه داده‌ی نوع int می‌باشد ولی «بدون علامت» است؛ یعنی اعداد منفی را در خود ذخیره نمی‌کند. این نوع داده می‌تواند یک عددصحیح در بازه‌ی ۰ تا ۶۵۵۳۵ را در خود ذخیره کند. مثال:

```
unsigned int Example = 9
```

word: این نوع داده، مشابه داده‌ی unsigned int است یعنی اعداد منفی را در خود ذخیره نمی‌کند و فقط می‌تواند یک عددصحیح در بازه‌ی ۰ تا ۶۵۵۳۵ را در خود ذخیره کند. مثال:

```
word Example = 10000
```

long: این نوع داده، مشابه داده int است ولی همان‌طور که از نام آن پیداست می‌تواند اعداد طولانی‌تری را در خود ذخیره کند. long می‌تواند یک عددصحیح

همان‌طور که مشاهده می‌کنید دو پرانتز () بعد از نام تابع آمده است. پارامترهای تابع در داخل پرانتز می آیند: پارامترهای تابع (Parameter1 و Parameter2) مقادیری هستند که می‌خواهیم به داخل تابع فرستاده شوند. اگر تابع دارای هیچ پارامتری نباشد، داخل پرانتزها خالی می‌ماند و اگر تابع بیش از یک پارامتر داشته باشد پارامترها با کاما و اسپیس از هم جدا می‌شوند.
بدنه‌ی تابع، در بین دو علامت آکولاد {} می‌آید. بدنه‌ی تابع شامل مجموعه‌ای از دستورات است که در توضیح بالا با عبارت function code نشان داده شده اند. در پایان این دستورات، می‌تواند دستور دیگری به نام return بیاید که نتیجه‌ی نهایی را به تابع دیگری که آنرا فراخوانده است برمی‌گرداند. توجه داشته باشید که انتهای هر دستور باید با علامت (;) مشخص شود.
در مثال زیر، یک تابع تعریف شده است که مقادیر x و y را می‌گیرد و آن‌ها را در هم ضرب می‌کند و نتیجه را برمی‌گرداند:

```
int myMultiply(int x, int y)
{
 int result;
 result = x * y;
 return result;
}
```

در مثال بالا، «نام» تابع myMultiply است که البته هر نام دلخواه دیگری نیز می‌توان به‌جای آن انتخاب کرد. داده‌هایی که تابع برمی‌گرداند از «نوع» int (عددصحیح) هستند. «پارامترهای تابع» را با عناوین دلخواه x و y نشان داده‌ایم و نوع آن‌ها را نیز int (عددصحیح) تعریف کرده‌ایم. بدنه‌ی تابع در بین {} آمده است و شامل سه خط دستوری به شرح زیر است:

```
int result;
result = x * y;
return result;
```

ابتدا کلمه‌ی دلخواه result تعریف شده است و نوع آن int (عددصحیح) انتخاب شده است. سپس حاصل ضرب x و y را در result قرار داده‌ایم. در انتها، مقدار result

را برگردانده‌ایم تا در جای دیگری از برنامه استفاده شود.
به‌طورکلی، تابع‌ها با نظم بخشیدن به دستورات، خوانایی برنامه را افزایش می‌دهند. همچنین اگر بخواهیم مجموعه‌ای از دستورات را در بخش‌های مختلف برنامه تکرار کنیم، در وقت ما صرفه‌جویی می‌کنند. در واقع، ما به‌جای نوشتن چندباره مجموعه‌ای از خطوط دستوری، ابتدا آن‌ها را در قالب یک تابع، تعریف کرده و سپس تابع را در بخش‌های مورد نیاز برنامه، فراخوانی می‌کنیم. در متن هر برنامه‌ی آردوینو، دو تابع ثابت و اصلی به نام‌های setup() و loop() وجود دارند که اکنون به شرح آن‌ها پرداخته می‌شود.

setup() و loop()

ساده‌ترین برنامه‌یا اسکچی که می‌توان در محیط آردوینو نوشت به این صورت است:

```
void setup()
{
//put your setup code here to run once
}

void loop()
{
//put your main code here to run repeatedly
}
```

نرم‌افزار آردوینو، فرمان‌های نوشته شده در برنامه را خط‌به‌خط می‌خواند و پایین می‌رود. در نگاه اول به متن این برنامه، سه کلمه‌ی void و setup و loop و نیز علایم () و {} و // جلب توجه می‌کنند که اکنون به شرح و یادآوری آن‌ها پرداخته می‌شود. هر برنامه‌ی ساده یا پیچیده‌ی آردوینو، باید دارای دو تابع ویژه به نام‌های setup() و loop() باشد. کار مشخصی که تابع setup() انجام می‌دهد آن است که: دستوری که برایش تعریف می‌کنیم را «فقط یک‌بار انجام می‌دهد». کار مشخصی که تابع loop() انجام می‌دهد آن است که: دستوری که برایش تعریف می‌کنیم را «آن‌قدر تکرار می‌کند تا بگوییم کافی است». همان‌طور که قبلاً گفته شد، دو پرانتز که به‌صورت () بعد از کلمات setup و loop آمده است یعنی

تابع ()setup دستورات داخل { } را فقط یک‌بار اجرا می‌کند ولی تابع ()loop دستورات داخل { } را نه فقط یک‌بار، بلکه آن‌قدر تکرار می‌کند تا بگوییم کافی است (نمودار ۵.۱).

نمودار ۵.۱: تابع ()setup دستورات داخل { } را فقط یک‌بار اجرا می‌کند ولی تابع ()loop دستورات داخل { } را به دفعات نامحدود تکرار می‌کند.

اکنون می‌خواهیم این برنامه‌ی ساده (که در بالا به آن اشاره شد) را در محیط آردوینو بنویسیم و آن را اجراء کنیم:

بورد آردوینو را به رایانه متصل کنید و نرم‌افزار آردوینو را اجراء نمایید. سپس وارد نرم‌افزار شده و مطابق با شکل ۵.۱ که در ابتدای فصل به آن اشاره شد، دکمه‌ی «ایجاد اسکچ جدید» را فشار دهید. با فشردن این دکمه، یک «ویرایشگر متن» جدید باز می‌شود. اگرچه در داخل این پنجره‌ی جدید، به‌صورت پیش‌فرض، همان اسکچ مورد نظر ما وجود دارد ولی بهتر است تمام متن موجود را پاک کرده و مجدداً خودتان آن را بنویسید تا نگارش کلمات، بهتر در ذهن‌تان ماندگار شود. زمانی‌که نگارش متن به پایان رسید، دکمه‌ی «ذخیره» را فشار دهید و آن را در محلی بر روی رایانه‌ی خود ذخیره کنید. اکنون دکمه‌ی «تایید» را فشار دهید تا اسکچ شما تایید و کامپایل شود. کامپایل یعنی ترجمه‌ی دستوراتی که شما نوشته اید به زبان رایانه. اگر متن شما بدون غلط نوشته شده باشد چند ثانیه طول می‌کشد تا عملیات تایید و ترجمه انجام شود. در خلال این پردازش، دو عبارت زیر در بخش «پنجره‌ی پیغام‌ها» ظاهر می‌شوند:

```
Compiling sketh...
Done compiling.
```

این عبارات توضیحی در یادآوری ساختار برنامه به ما بسیار موثرند. این عبارات توضیحی، تاثیری در اجرای دستورات برنامه ندارند و تنها به‌عنوان یادداشت‌هایی دلخواه برای خودمان محسوب می‌شوند و بهتر است که نگاشته شوند هرچند اجباری نمی‌باشند. محیط برنامه‌نویسی آردوینو، عبارات توضیحی را به‌صورت خودکار، به رنگ خاکستری نمایش می‌دهد تا از عبارات دستوری متمایز شوند. عبارات توضیحی می‌توانند به زبان دلخواه، مثلاً به زبان فارسی، نوشته شوند:

```
void setup()
{
//کدها را این‌جا بنویسید تا یک‌بار اجراء شوند
}
```

عبارات توضیحی به دو گروه تقسیم‌بندی می‌شوند:

الف. تک‌خطی

ب. چندخطی

الف. عبارات توضیحی تک‌خطی: همان‌طور که از عنوان آن استنباط می‌شود، توضیحات تک‌خطی باید در یک خط خلاصه شوند. شروع توضیحات تک‌خطی می‌تواند از ابتدا و یا وسط خط (یعنی پس از بیان دستور) باشد. عبارت توضیحی با علامت // شروع شده و تا انتهای همان خط ادامه می‌یابد. مثال:

```
void setup()
{
//دستورات این بخش فقط یک‌بار انجام می‌شوند
}
void loop() //دستورات بین آکولادهای زیر تکرار می‌شوند
{
}
```

ب. عبارات توضیحی چندخطی: این نوع عبارات در بین دو علامت /* و */ قرارمی‌گیرند و می‌توانند چندین خط را به خود اختصاص دهند. مثال:

```
/*
Setup & Loop example
مثال برای ست‌آپ و لوپ
کدها را بین آکولادها { } قرار دهید
تاریخ ویرایش: دی ماه
This example code is in the public domain
*/

void setup()
{
//put your setup code here to run once
}

void loop()
{
//put your main code here to run repeatedly
}
```

در داخل متن عبارات چندخطی، می‌توان از عبارت تک‌خطی نیز استفاده کرد:

```
/*
Setup & Loop example
مثال ست‌آپ و لوپ
کدها را بین آکولادها { } قرار دهید
تاریخ ویرایش: دی ماه
This example code is in the public domain
//single line comment is OK inside a multi-line
//  عبارت تک‌خطی در متن یک عبارت توضیحی چندخطی
*/
```

یک‌بار دیگر مثال فوق را با توضیحات کامل چندخطی و تک خطی در زیر می‌آوریم و برای خوانایی بیشتر و درک بهتر دستورات، از فاصله‌های خالی (اسپیس و اینتر) در بین خطوط و کلمات استفاده می‌کنیم و آکولاد باز { را نیز در کنار تابع قرار می‌دهیم:

```
/*
Setup & Loop example
مثال برای ست‌آپ و لوپ
کدها را بین آکولادها { } قرار دهید
تاریخ ویرایش: دی ماه
This example code is in the public domain
*/

void setup() {
 // put your setup code here, to run once:

}

void loop() {
 // put your main code here, to run repeatedly:

}
```

توجه نمایید که کلیه‌ی قسمت‌هایی که در اسکچ فوق با سایه‌ی خاکستری مشخص شده اند، تاثیری در روند اجرای برنامه ندارند و قابل حذف اند. همچنین کلیه‌ی فاصله‌ها و خطوط خالی (به غیر از فاصله‌ی بین void با setup و loop) می‌توانند حذف شوند:

```
void setup(){}void loop(){}
```

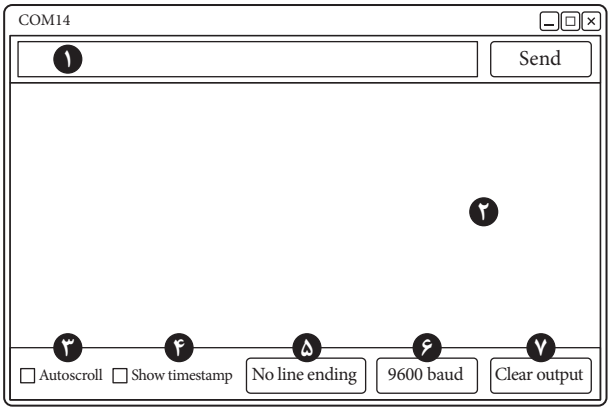
اگرچه کدهای نوشته شده به‌صورت فوق، بدون مشکل اجراء می‌شوند ولی با توجه به آن‌که در اسکچ‌های پیچیده، موجب دشواری در خواندن و درک دستورات می‌گردند لذا توصیه می‌شود که در زمان نگارش کدها، از نظم و ترتیب مشخصی پیروی کنید.

خروج داده‌ها از نمایشگر سریال

یکی از راه‌های تعامل بین کاربر و آردوینو، استفاده از «نمایشگر سریال»[1] است (شکل ۵.۲). از طریق نمایشگر سریال، شما می‌توانید اطلاعات متنی را برای آردوینو بنویسید و یا اطلاعات متنی نوشته‌شده توسط آردوینو را بخوانید. فرض کنید که شما یک حسگر دما دارید و می‌خواهید هر پنج ثانیه یک‌بار از تغییرات دما

۱. Serial Monitor

مطلع شوید. به این منظور می‌توانید درجه‌ی حرارت را، به‌صورت اعدادی که هر پنج ثانیه یک‌بار تغییر می‌کنند، از روی نمایشگر سریال بخوانید.
برای دسترسی به پنجره‌ی «نمایشگر سریال»، ابتدا بورد آردوینو را به رایانه‌ی خود وصل کنید و سپس نرم‌افزار آردوینو را اجرا نموده و بر روی آیکون «نمایشگر سریال» در بالای صفحه اصلی کلیک کنید. توجه نمایید که بدون اتصال بورد به رایانه، پنجره‌ی نمایشگر سریال باز نمی‌شود.



شکل ۵.۲: پنجره‌ی «نمایشگر سریال»

در پنجره‌ی «نمایشگر سریال»، پنج بخش مجزا به شرح زیر وجود دارد:
۱. محل ورود اطلاعات متنی به برنامه (توسط کاربر): در برخی از برنامه‌ها می‌توانیم اطلاعاتی را از کاربر بگیریم تا برنامه برمبنای آن اطلاعات، تصمیم‌گیری کند. مثلاً در یک برنامه‌ی تنظیم دمای اتاق، می‌توانید از کاربر بخواهید که مقدار دمای مورد نظرش را در این‌جا بنویسد.
۲. محل نمایش اطلاعات خروجی برنامه: اطلاعات خروجی برنامه، در قالب متن، در این بخش نمایش داده می‌شوند. مثلاً: اکنون دمای اتاق به ۱۹ درجه‌ی سانتیگراد رسید.
۳. چنان‌چه حجم داده‌های نمایش داده شده بر روی پنجره، بیشتر از یک صفحه باشد، به‌صورت خودکار آن‌قدر صفحه را بالا می‌برد تا پایین صفحه که حاوی اطلاعات جدید است در معرض نمایش قرارگیرد.
۴. با فعال کردن این گزینه، زمان (ساعت) در ابتدای هر خط از داده‌های خروجی درج می‌شود.

```
void setup() {
 Serial.begin(9600);
 Serial.print("Hello");
}

void loop() {
}
```

اکنـون، مثـال فـوق را اندکـی تغییـر می‌دهیـم تـا عبـارت Hello بـه دفعـات نامحـدود بـر روی نمایشـگر سـریال نوشـته شـود. بـرای ایـن کار، دسـتور اول را در بدنـه‌ی تابـع setup() نـگاه می‌داریـم، ولـی دسـتور دوم را بـه بدنـه‌ی تابـع loop() منتقـل می‌کنیـم تـا بـه دفعـات نامحـدود تکـرار شـود:

```
void setup() {
 Serial.begin(9600);
}

void loop() {
 Serial.print("Hello");
}
```

خروجی این برنامه به‌صورت زیر است:

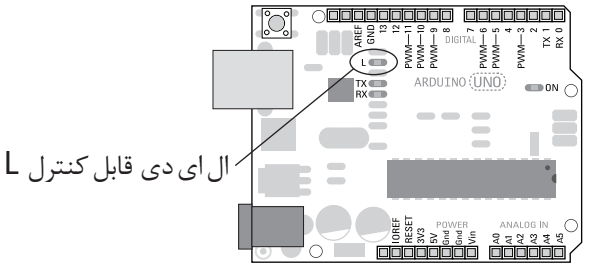
HelloHelloHello...

تابعـی بـه نـام Serial.println() وجـود دارد که مشـابه تابـع Serial.print() اسـت، بـا این تفاوت کـه بعـد از نوشـتن خروجی، بـه ابتدای خط بعـد می‌رود. یعنی به‌جـای آن‌که عبـارات خروجی، پشـت سـر هـم بیایند و به هم بچسـبند، هر کـدام در ابتدای یک خط جدید نوشـته می‌شـوند:

```
void setup() {
 Serial.begin(9600);
}

void loop() {
 Serial.println("Hello");
}
```

کنترل روشن و خاموش‌شدن ال‌ای‌دی

یکـی دیگـر از راه‌هـای تعامل بین کاربر و آردوینو، اسـتفاده از «ال‌ای‌دی» قابل‌کنترلی اسـت کـه بـر روی بورد آردوینو وجود دارد و با حرف L مشـخص شـده اسـت (شـکل ۵.۳). در این پـروژه می‌خواهیـم بـا آن ارتباط برقرار کرده و روشـن و خاموش‌شـدن آن را کنتـرل نماییم.



شکل ۵.۳: محل قرارگیری ال‌ای‌دی L بر روی بورد آردوینو

بـورد آردوینـو را بـه رایانـه متصـل کنیـد و نرم‌افـزار آردوینـو را اجـرا نماییـد. سـپس یـک فایـل جدیـد ایجـاد نمـوده و دسـتورات زیـر را داخـل آن بنویسـید:

```
void setup() {
 pinMode(13, OUTPUT);
}

void loop() {
 digitalWrite(13, HIGH);
 delay(1000);
 digitalWrite(13, LOW);
 delay(1000);
}
```

دکمه‌هـای «تاییـد» و «بارگـذاری» را فشـار دهیـد. اگـر از نـگارش صحیـح کدهـا مطمئـن هسـتید می‌توانیـد فقـط دکمـه‌ی «بارگـذاری» را فشـار دهیـد. اکنـون مشـاهده می‌کنیـد کـه ال‌ای‌دی L (موجـود بـر روی بـورد آردوینـو) شـروع به روشـن و خاموش‌شـدن می‌کند. مـدت زمـان روشـن بـودن آن ۱ ثانیـه و مـدت زمـان خاموش بـودن آن نیـز ۱ ثانیه اسـت. در این پروژه با چند دستور و واژه‌ی جدید به شرح زیر مواجه هستید:

pinMode()
OUTPUT
digitalWrite()
HIGH
LOW
delay()

اکنون به شرح تک‌تک این دستورات و واژگان می‌پردازیم.

pinMode()

همان‌طور که قبلاً اشاره شد، درگاه‌های دیجیتال موجود بر روی بورد آردوینو می‌توانند هم به‌عنوان ورودی و هم به‌عنوان خروجی استفاده شوند. با استفاده از تابع pinMode() می‌توان نوع ورودی یا خروجی بودن این درگاه‌ها را تنظیم نمود. ساختار این تابع به‌صورت زیر است:

```
pinMode(pin, mode)
```

در بخش pin شماره‌ی درگاه موردنظر خود را می‌نویسیم. در بخش mode برای آردوینو مشخص می‌کنیم که می‌خواهیم آن درگاه، ورودی یا خروجی (INPUT یا OUTPUT) باشد. مثلاً اگر بخواهیم درگاه شماره ۱۳ به‌عنوان درگاه خروجی تنظیم شود از دستور زیر استفاده می‌کنیم:

```
pinMode(13, OUTPUT)
```

در مورد ال‌ای‌دی L (موجود بر روی بورد آردوینو) ذکر این نکته حائز اهمیت است که این ال‌ای‌دی مستقیماً به درگاه شماره‌ی ۱۳ متصل است. بنابراین اگر در تابع pinMode() درگاه شماره ۱۳ را خروجی تعریف کنیم، می‌توان از طریق همین درگاه ۱۳، کنترل ال‌ای‌دی را در دست گرفت. همچنین این امکان وجود دارد که به‌جای استفاده از درگاه ۱۳، آدرس‌دهی را مستقیماً و با عنوان LED_BUILTIN انجام دهیم:

```
pinMode(LED_BUILTIN, OUTPUT)
```

در واقع در این بخش از اسکچ، به آردوینو می‌گوییم که ال‌ای‌دی موجود بر روی بورد آردوینو را به‌عنوان یک خروجی بشناسد و سیگنالی را جهت روشن‌کردن آن بفرستد.

- HIGH برای روشن‌کردن ال‌ای‌دی
- LOW برای خاموش‌کردن ال‌ای‌دی

توجه داشته باشید که تابع digitalWrite() یک کنترل‌کننده‌ی دیجیتال است (آنالوگ نیست) و باید بر روی درگاه‌های دیجیتال بورد آردوینو اعمال شود ولی در بورد آردوینو (مدل اونو)، درگاه‌های ورودی آنالوگ نیز می‌توانند به‌عنوان درگاه دیجیتال استفاده شوند. همچنین این تابع فقط می‌تواند دو حالت داشته باشد: روشن یا خاموش (HIGH یا LOW). در بورد آردوینو (مدل اونو) اگر تابع digitalWrite() را در حالت HIGH تنظیم کنیم، ولتاژ خروجی آن ۵ ولت می‌شود و اگر آن را در حالت LOW قرار دهیم این مقدار به صفر می‌رسد. این ولتاژ ۰ یا ۵ ولت، به درگاهی ارسال می‌شود که شماره‌ی آن را در قسمت pin به آردوینو می‌گوییم: مثلاً درگاه شماره‌ی ۱۳. در مثال حاضر، دو بار از تابع digitalWrite()استفاده کرده‌ایم:
الف. یک‌بار برای آن‌که به آردوینو بگوییم درگاه شماره‌ی ۱۳ را روشن کند:

```
digitalWrite(13, HIGH)
```

ب. بار دیگر برای آن‌که به آردوینو بگوییم درگاه شماره‌ی ۱۳ را خاموش کند:

```
digitalWrite(13, LOW)
```

با توجه به آن‌که این دو دستور در تابع loop() قرارگرفته اند بنابراین به دفعات نامحدود تکرار می‌شوند. یعنی ال‌ای‌دی به‌صورت مداوم روشن و سپس خاموش می‌شود یا اصطلاحاً چشمک می‌زند. در این مثال، زمان چشمک‌زدن ال‌ای‌دی ۲ ثانیه است: ۱ ثانیه روشن است و ۱ ثانیه خاموش است. برای تعیین مدت زمان روشن و خاموش بودن آن، از تابعی به نام delay() استفاده کرده ایم که اکنون به شرح آن می‌پردازیم.

delay()

برای تنظیم مدت زمان روشن و خاموش بودن ال‌ای‌دی می‌توانیم از تابع delay() استفاده کنیم. این تابع، دستوری که قبل از آن اجرا شده است را به مدت زمان دلخواه ما، در حال اجرا نگه می‌دارد. یعنی آن دستور اجرا می‌شود و اجرای آن تا زمانی‌که ما تعیین کرده ایم ادامه می‌یابد. پس از پایان زمان تعیین شده، به‌صورت

معمـول، خـط بعـدی برنامـه اجـرا می‌شـود. سـاختار ایـن تابـع به‌صـورت زیـر اسـت:

```
delay(time)
```

در بخـش time مـدت زمانـی را کـه می‌خواهیـم برنامـه‌ی مـا معطـل بمانـد، تعییـن می‌کنیـم. ایـن زمـان برحسـب میلی‌ثانیـه (هزارم‌ثانیـه) نوشـته می‌شـود. مثـلاً می‌خواهیـم بـه آردوینـو بگوییـم کـه پـس از روشـن‌کردن ال‌ای‌دی، یـک ثانیـه معطـل بمانـد و سـپس آن را خامـوش کنـد:

- روشن‌کردن: digitalWrite(13, HIGH)
- توقف ۱ ثانیه: delay(1000)
- خاموش‌کردن: digitalWrite(13, LOW)

یا مثلاً بعد از خاموش‌کردن ال‌ای‌دی، دو ثانیه صبر کند و سپس آن را روش کند:

- خاموش‌کردن: digitalWrite(13, LOW)
- توقف ۲ ثانیه: delay(2000)
- روشن‌کردن: digitalWrite(13, HIGH)

بـه مثـال اصلـی بازمی‌گردیـم: می‌خواهیـم ال‌ای‌دی روشـن شـود؛ یـک ثانیـه روشـن بمانـد؛ سـپس خامـوش شـود؛ و یک ثانیه خامـوش بماند. بنابراین دسـتور را به‌صورت زیر می‌نوسـیم:

- روشن‌کردن: digitalWrite(13, HIGH)
- توقف ۱ ثانیه: delay(1000)
- خاموش‌کردن: digitalWrite(13, LOW)
- توقف ۱ ثانیه: delay(1000)

بـرای آن‌کـه ایـن دسـتورات بـه دفعـات نامحـدود تکـرار شـوند و ال‌ای‌دی به‌صـورت مداوم روشـن و خامـوش شـود، مجموعـه ی ایـن دسـتورات را در تابـع ()loop قـرار می‌دهیـم:

```
void loop() {
 digitalWrite(13, HIGH);
 delay(1000);
 digitalWrite(13, LOW);
 delay(1000);
}
```

اکنون متن اسکچ را به‌صورت زیر تغییر می‌دهیم:

```
const int LED = 13;
void setup() {
 pinMode(LED, OUTPUT);
}

void loop() {
 digitalWrite(LED, HIGH);
 delay(1000);
 digitalWrite(LED, LOW);
 delay(1000);
}
```

در اسکچ بالا، عدد ۱۳ یک عدد معنادار برای آردوینو است زیرا شماره‌ی درگاهی را نشان می‌دهد که ال‌ای‌دی L (موجود بر روی بورد آردوینو) به آن متصل است. یعنی اگر بخواهیم ال‌ای‌دی L روشن شود باید سیگنال را به درگاه شماره‌ی ۱۳ بفرستیم. اگرچه عدد ۱۳ برای آردوینو معنادار است ولی برای انسان معنای مشخصی ندارد. در مثال بالا خواسته ایم به‌جای نوشتن عدد ۱۳ (شماره‌ی درگاه ال‌ای‌دی)، از یک کلمه‌ی گویا و معنادار مثلاً از کلمه‌ی LED استفاده کنیم. اگرچه کلمه‌ی LED برای آردوینو معنا ندارد ولی برای انسان معنادار است. بنابراین در خط اول اسکچ، کلمه‌ی LED (یا هر کلمه‌ی دلخواه دیگری) را به‌عنوان یک متغیر به آردوینو معرفی می‌کنیم تا در خطوط بعدی برنامه در هر کجا که لازم بود به‌جای عدد ۱۳ از کلمه‌ی LED استفاده کنیم. پس «نام» متغیر ما LED است. سپس «نوع» متغیر را int انتخاب می‌کنیم. همان‌طور که قبلاً گفته شد متغیرهای int برای نگهداری اعداد صحیح مناسب اند. با توجه به آن‌که می‌خواهیم عدد مربوط به شماره‌ی درگاه ال‌ای‌دی (عدد ۱۳) به متغیر انتساب داده شود، بنابراین استفاده از int انتخاب مناسبی است. چون قرار نیست شماره‌ی درگاه ال‌ای‌دی در طول اجرای برنامه تغییر کند بنابراین عبارت const را نیز قبل از int می‌نویسیم. برای آن‌که مقدار ۱۳ به متغیر LED انتساب داده شود از علامت (=) استفاده می‌کنیم:

فصل ٦. پروژه‌های آردوینو

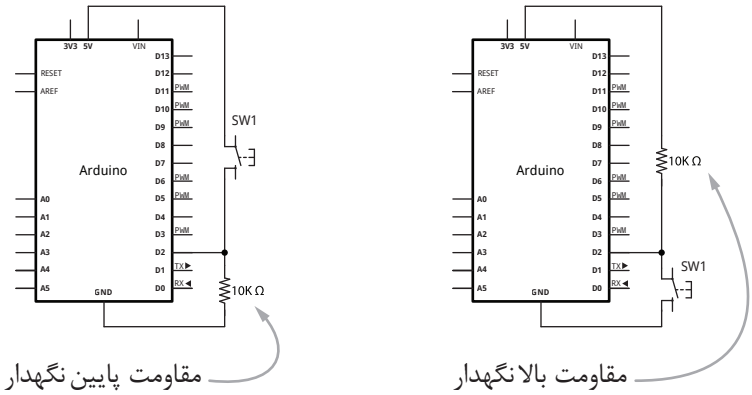
پس از یادگیری مطالب گفته‌شده در فصل‌های قبل، اکنون می‌توانید پروژه‌های زیر را به‌صورت عملی اجراء کنید تا درک بهتری از نرم‌افزار و سخت‌افزار آردوینو به‌دست آوردید. ترتیب معرفی پروژه‌ها از ساده به دشوار است و در صورت تمایل می‌توانید متن کد پروژه‌ها را از آدرس github.com/ArduinoKey دانلود نمایید.

پروژه‌ی اول: روشن و خاموش‌کردن ال‌ای‌دی توسط آردوینو

در این مثال می‌خواهیم روشن و خاموش‌شدن یک ال‌ای‌دی را با استفاده از آردوینو کنترل‌کنیم. قطعات مورد نیاز برای انجام این پروژه عبارت‌اند از:

بردبورد
سیم‌جامپر
مقاومت ۱ کیلواهم (قهوه ای، مشکی، قرمز)
ال‌ای‌دی
بورد آردوینو

همان‌طور که پیش‌تر گفته شد، هر ال‌ای‌دی دارای دو قطب آند (مثبت) و کاتد (منفی) است. پایه‌ی کاتد، کوتاه‌تر است و انتهای آن در داخل حباب، خم شده است. همچنین کناره‌ی حباب در سمت کاتد، کمی صاف است. پایه‌های ال‌ای‌دی را مستقیم و بدون واسطه به درگاه‌های بورد آردوینو متصل می‌کنیم. پایه‌ی آند ال‌ای‌دی به درگاه ۱۳ و پایه‌ی کاتد ال‌ای‌دی به درگاه GND بورد آردوینو وصل می‌شود. در شکل ۶.۱ نحوه‌ی اتصال ال‌ای‌دی به آردوینو نشان داده شده است.

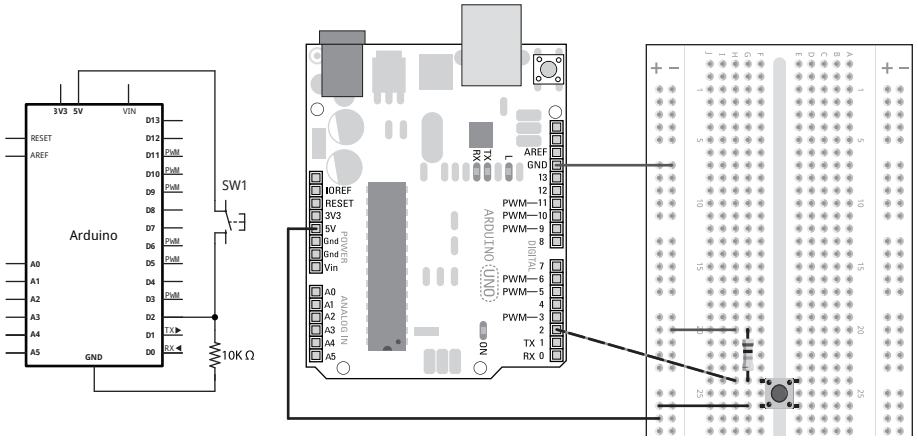


شکل ۶.۴: مقاومت‌های بالانگهدار و پایین‌نگهدار

در مدار بالانگهدار، زمانی‌که کلید آزاد است، مقدار HIGH توسط درگاه خوانده می‌شود ولی زمانی‌که کلید فشرده شود، درگاه به‌طور مستقیم به زمین وصل شده و بنابراین درگاه در وضعیت LOW قرار می‌گیرد.

در مدار پایین‌نگهدار، زمانی‌که کلید آزاد است، مقدار LOW توسط درگاه خوانده می‌شود ولی زمانی‌که کلید فشرده شود، درگاه به‌طور مستقیم به‌ی مثبت ۵ ولت آردوینو وصل شده و بنابراین درگاه در وضعیت HIGH قرار می‌گیرد.

توجه داشته باشید که عدم وجود مقاومت، سبب ایجاد اتصال کوتاه در مدار می‌شود. مقدار مقاومت بالانگهدار و پایین‌نگهدار مورد استفاده برای آردوینو، در محدوده‌ی ۱۰ کیلواهم است. در شکل ۶.۵ نحوه‌ی اتصال مقاومت پایین‌نگهدار خارجی، به پایه‌ی دیجیتال شماره ۲ آردوینو آمده است.



شکل ۶.۵: نحوه‌ی اتصال تک‌سوییچ و مقاومت به بورد آردوینو

برای اتصال تک‌سوییچ، هر چهار پایه‌ی آن باید وارد سوراخ‌های بردبورد شوند. لطفا به این نکته توجه داشته باشید که پایه‌هایی که از داخل تک‌سوییچ به هم وصل هستند باید بر روی سوراخ‌های عرضی بردبورد که از داخل بردبورد به هم وصل هستند نصب شوند و در غیر این صورت، باعث اتصال کوتاه بین دو ردیف می‌شوند. سپس بورد آردوینو را به رایانه متصل کرده و از طریق نرم‌افزار آردوینو، برنامه‌ی زیر را بر روی بورد آردوینو بارگذاری می‌کنیم:

```
const int buttonPin = 2;
const int ledPin = 13;
int buttonState = 0;

void setup() {
 pinMode(buttonPin, INPUT);
 pinMode(ledPin, OUTPUT);
}

void loop() {
 buttonState = digitalRead(buttonPin);
 if (buttonState == HIGH) {
  digitalWrite(ledPin, HIGH);
 }
 else {
  digitalWrite(ledPin, LOW);
 }
}
```

همان‌طور که در متن برنامه مشاهده می‌نمایید، ابتدا مقادیر ۲ و ۳ را به ثابت‌های buttonPin و ledPin نسبت می‌دهیم. مقادیر ۲ و ۳، معرف شماره‌ی درگاه‌های بورد آردوینو هستند که به‌ترتیب به کلید فشاری و ال‌ای‌دی موجود بر روی بورد آردوینو متصل اند. می‌توان این انتساب نام را انجام نداد و در خطوط بعدی برنامه، مستقیما از همین اعداد برای آدرس دهی کلید فشاری و ال‌ای‌دی استفاده کرد ولی بهتر است برای خوانایی و درک بهتر برنامه، به‌جای اعداد نا مفهوم از اسامی معنادرا استفاده کنیم. متغیری به نام buttonState را از نوع عددصحیح (int) تعریف می‌کنیم که وضعیت HIGH یا LOW کلید فشاری در آن ذخیره خواهد شد. در بخش ()setup، وضعیت ورودی و خروجی درگاه‌های شماره ۲ و ۱۳ را تعیین می‌کنیم. در بخش ()loop وضعیت buttonPin را توسط تابع ()digitalRead می‌خوانیم و در متغیر buttonState قرار می‌دهیم. اگر مقدار buttonState در وضعیت HIGH باشد درگاه متصل به ال‌ای‌دی (درگاه شماره ۱۳) نیز HIGH می‌شود و در غیر این صورت LOW می‌گرددو ال‌ای‌دی خاموش می‌شود. توجه داشته باشید که وقتی کلید فشاری آزاد است، درگاه ۲ از طریق مقاومت پایین‌نگهدار در وضعیت LOW قرار دارد و وقتی که کلید فشزده شود، ارتباط درگاه ۲ با مثبت ۵ ولت برقرار شده و در وضعیت HIGH قرار می‌گیرد. وضعیت خاموش و روشن‌شدن ال‌ای‌دی نیز از وضعیت کلید تبعیت خواهد کرد. پس از آشنایی با مقاومت پایین‌نگهدار، اکنون به بررسی مقاومت بالانگهدار می‌پردازیم. برای اتصال مقاومت بالانگهدار به بورد آردوینو، با دو گزینه روبرو هستیم:

- مقاومت بالانگهدار خارجی
- مقاومت بالانگهدار داخلی

واضح است که در حالت اول، یک قطعه مقاومت، بین درگاه مورد نظر و درگاه‌ی مثبت ۵ ولت آردوینو وصل می‌شود. نکته ی حایز اهمیت آن است که میکروکنترلر آردوینو در داخل خود دارای مقاومت بالانگهدار است که می‌توانیم آن را از طریق برنامه نویسی فعال کنیم که اکنون به شرح آن می‌پردازیم. پیش‌تر با تابع ()pinMode آشنا شده اید و می‌دانید که ساختار آن به‌صورت زیر است:

```
pinMode(pin, mode)
```

در بخش pin شماره‌ی درگاه مورد نظر خود را می‌نویسیم و در بخش mode برای

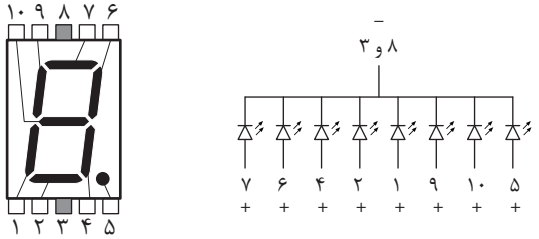
را نیـاز دارد. بـرای حـل ایـن مشـکل از تابـع ()map اسـتفاده می‌کنیـم. همان‌طـور کـه در بخـش برنامه‌نویسـی در محیـط آردوینـو توضیـح دادیـم، از تابـع ()map بـرای نگاشـت بیـن اعضـای دو مجموعـه اسـتفاده می‌شـود. مثـلاً اگـر بخواهیـم اعـداد موجـود در بـازه‌ی ۰ تـا ۱۰۲۳ را بـه بـازه‌ی ۰ تـا ۱۸۰ تبدیـل کنیـم از تابـع نگاشـت اسـتفاده می‌کنیـم.

پروژه‌ی هشتم: کنترل سون‌سگمنت توسط آردوینو

در ایـن پـروژه می‌خواهیـم یـک سون‌سـگمنت را توسـط آردوینـو کنتـرل نماییـم. ابتـدا اعـداد ۰ تـا ۹ بـا فاصلـه‌ی زمانـی ۱ ثانیـه بـر روی سون‌سـگمنت نمایـش داده خواهنـد شـد. پـس از آن، اعـداد ۹ تـا ۰ بـا فاصلـه‌ی زمانـی ۰/۲۵ ثانیـه نمایـش داده خواهنـد شـد. قطعـات مـورد نیـاز بـرای انجـام ایـن پـروژه عبارت‌انـد از:

بردبورد
سیم‌جامپر
مقاومت ۱ کیلواهم (قهوه ای، مشکی، قرمز)
سون‌سگمنت (تکی، ۱۳×۱۹ میلی متر، کاتدمشترک)
بورد آردوینو

همان‌طـور کـه پیش‌تـر گفتـه شـد، سون‌سـگمنت از هفـت ال‌ای‌دی مجـزا تشـکیل شـده کـه در کنـار یکدیگـر قرارگرفتـه انـد و تک‌به‌تـک قابـل کنتـرل انـد و می‌تـوان از آن‌هـا بـرای نمایـش اعـداد ۰ تـا ۹ اسـتفاده کـرد. در مثـال حاضـر، از یـک سون‌سـگمنت تکـی ۱۳×۱۹ میلی‌متـری اسـتفاده خواهیـم کـرد. ایـن سون‌سـگمنت‌ها بـه دو گـروه کاتدمشـترک و آندمشـترک تقسـیم می‌شـوند. کاتدمشـترک یعنـی پایـه‌ی کاتـد تمـام ال‌ای‌دی‌هـای موجـود در سون‌سـگمنت بـه یکدیگـر متصـل هسـتند و هـر ال‌ای‌دی از پایـه‌ی آنـد اختصاصـی خـود تغذیـه و کنتـرل می‌شـود، درحالی‌کـه آندمشـترک یعنـی پایـه‌ی آنـد تمـام ال‌ای‌دی‌هـای موجـود در سون‌سـگمنت بـه یکدیگـر متصـل هسـتند و هـر ال‌ای‌دی از پایـه‌ی کاتـد اختصاصـی خـود تغذیـه و کنتـرل می‌گـردد. سون‌سـگمنت مـورد اسـتفاده در مثـال حاضـر، از نـوع کاتدمشـترک اسـت کـه در شـکل ۶.۱۶ بـه صـورت نمادیـن نمایـش داده شـده اسـت.

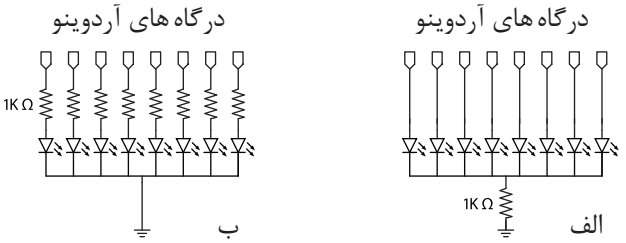


شکل ۶.۱۶: سون‌سگمنت کاتدمشترک

همان‌طور که در شکل ۶.۱۶ مشاهده می‌کنید، این سون‌سگمنت دارای ۱۰ پایه به شرح زیر است:

- دو پایه برای کاتدمشترک (پایه‌های ۳ و ۸)
- هفت پایه برای تغذیه‌ی هفت ال‌ای‌دی (پایه‌های ۱، ۲، ۴، ۶، ۷، ۹ و ۱۰)
- یک پایه برای تغذیه‌ی ممیز (پایه ۵)

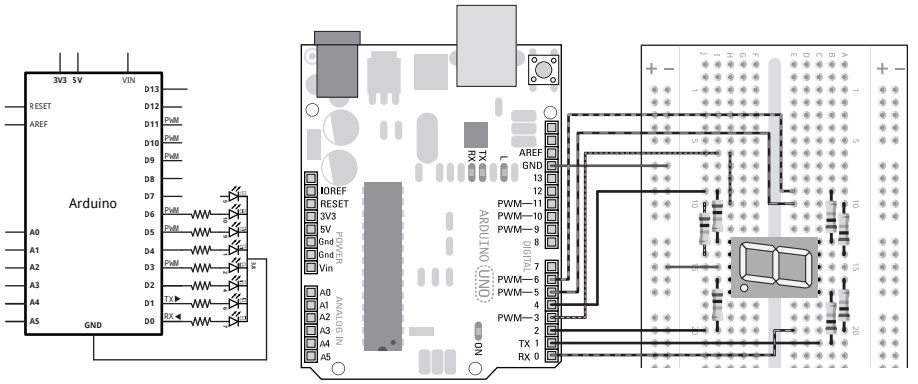
یک سون‌سگمنت قرمز معمولی، مشابه با نمونه‌ی مورد استفاده در این مثال، معمولاً با ۲ ولت و ۲۰ میلی‌آمپر راه‌اندازی می‌شود. به منظور حفاظت از آن در برابر جریان اضافی، می‌توان به یکی از دو حالتی که در شکل ۶.۱۷ نمایش داده شده است، پایه‌ها را به مقاومت‌های محافظ متصل نمود.



شکل ۶.۱۷: حفاظت از سون‌سگمنت در برابر جریان اضافی: الف. می‌توان تنها یک مقاومت در مسیر پایه‌ی مشترک قرار داد؛ ب. می‌توان برای هریک از پایه‌های مستقل، یک مقاومت قرار داد.

همان‌طور که در شکل ۶.۱۷ مشاهده می‌کنید، برای حفاظت از سون‌سگمنت در برابر جریان اضافی می‌توان تنها یک مقاومت در مسیر پایه‌ی مشترک قرار داد و یا برای هریک از پایه‌های مستقل، یک مقاومت در نظر گرفت. اگرچه در هر دو حالت فوق، از سون‌سگمنت در برابر جریان اضافی محافظت خواهد شد ولی استفاده از یک مقاومت

در مسیر پایه‌ی مشترک، می‌تواند سبب تفاوت در میزان نوردهی خطوط سون‌سگمنت گردد، به خصوص زمانی‌که تعداد زیادی از آن‌ها روشن باشند. بنابراین بهتر است که برای هریک از پایه‌های مستقل، از یک مقاومت جداگانه استفاده شود (حالت ب شکل ۶.۱۷). اکنون با توجه به توضیحات فوق، قطعات را مطابق شکل ۶.۱۸ به یکدیگر متصل کنید.



شکل ۶.۱۸: نحوه‌ی اتصال سون‌سگمنت و مقاومت‌های محافظ به بورد آردوینو

ترتیب اتصال پایه‌های سون‌سگمنت به درگاه‌های بورد آردوینو، در جدول ۶.۱ آمده است. با توجه به آن‌که سون‌سگمنت به‌کاررفته در این پروژه، از نوع کاتدمشترک است، بنابراین پایه‌های ۳ و ۸ آن به درگاه GND بورد آردوینو وصل می‌شوند. در مدار حاضر، پایه ۵ سون‌سگمنت (ممیز) آزاد است و مورد استفاده قرار نمی‌گیرد.

جدول ۶.۱: ترتیب اتصال پایه‌های سون‌سگمنت به درگاه‌های بورد آردوینو

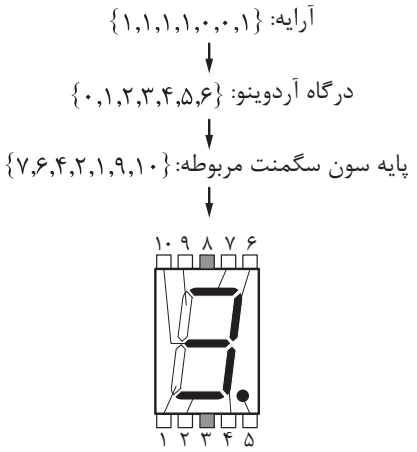
شماره‌ی درگاه آردوینو	شماره‌ی پایه‌ی سون‌سگمنت
۰	۷
۱	۶
۲	۴
۳	۲
۴	۱
۵	۹
۶	۱۰

اکنـون از طریـق نرم‌افـزار آردوینـو، برنامـه‌ی زیر را بـر روی بـورد آردوینو بارگـذاری می‌کنیم.

```
int NUM [10][7] = {
 { 1,1,1,1,1,1,0 },
 { 0,1,1,0,0,0,0 },
 { 1,1,0,1,1,0,1 },
 { 1,1,1,1,0,0,1 },
 { 0,1,1,0,0,1,1 },
 { 1,0,1,1,0,1,1 },
 { 1,0,1,1,1,1,1 },
 { 1,1,1,0,0,0,0 },
 { 1,1,1,1,1,1,1 },
 { 1,1,1,1,0,1,1 }
};

void setup() {
 pinMode(0, OUTPUT);
 pinMode(1, OUTPUT);
 pinMode(2, OUTPUT);
 pinMode(3, OUTPUT);
 pinMode(4, OUTPUT);
 pinMode(5, OUTPUT);
 pinMode(6, OUTPUT);
}

void loop() {
 for(int i=0;i<10;i++) {
 for(int j=0;j<7;j++) {
  digitalWrite(j,NUM[i][j]);
 }
 delay(1000);
 }

 for(int i=9;i>0;i--) {
 for(int j=7;j>=0;j--) {
  digitalWrite(j,NUM[i][j]);
 }
 delay(250);
 }
}
```

گفتیم که سون‌سگمنت از هفت ال‌ای‌دی یا خط نوری تشکیل می‌شود و به واسطه روشن یا خاموش ماندن این خطوط، اعداد مختلف بر صفحه سون‌سگمنت نقش می‌بندند. همان‌طور که در ابتدای اسکچ فوق مشاهده می‌کنید، برای تعریف وضعیت روشن یا خاموش بودن خطوط سون‌سگمنت، از آرایه یا Array استفاده کرده ایم. پیش‌تر در بخش برنامه‌نویسی در محیط آردوینو، با مفهوم آرایه‌ها آشنا شده اید. ابتدا آرایه‌ای از اعداد صحیح را با نام دلخواه NUM تشکیل می‌دهیم. اعداد موجود در این آرایه بیان‌گر وضعیت روشن (۱) یا خاموش (۰) بودن خطوط سون‌سگمنت می‌باشند. مثلاً آرایه‌ی {۱,۱,۱,۱,۰,۰,۱} می‌گوید کدام خطوط از سون‌سگمنت باید روشن شوند تا عدد ۳ نمایش داده شود. این اعداد از چپ به راست به‌ترتیب به درگاه‌های دیجیتال شماره ۰ تا ۶ بورد آردوینو و پایه‌های ۷، ۶، ۴، ۲، ۱، ۹ و ۱۰ سون‌سگمنت مربوط می‌شوند. همان‌طور که به یاد می آورید، ترتیب اتصال پایه‌های سون‌سگمنت به درگاه‌های بورد آردوینو را خودمان در زمان بستن مدار تعریف کردیم. رابطه‌ی اعداد آرایه با پایه‌های سون‌سگمنت در شکل ۶.۱۹ نشان داده شده است.

- آرایه: { ۱,۱,۱,۱,۰,۰,۱ } (از چپ به راست، صفر خاموش، یک روشن)
- شماره درگاه مربوطه بر روی بورد آردوینو: {۰,۱,۲,۳,۴,۵,۶}
- شماره پایه مربوطه بر روی سون‌سگمنت: {۷,۶,۴,۲,۱,۹,۱۰} (خطی که باید بر روی سون‌سگمنت روشن شود)



شکل ۶.۱۹: رابطه‌ی اعداد آرایه با پایه‌های سون‌سگمنت

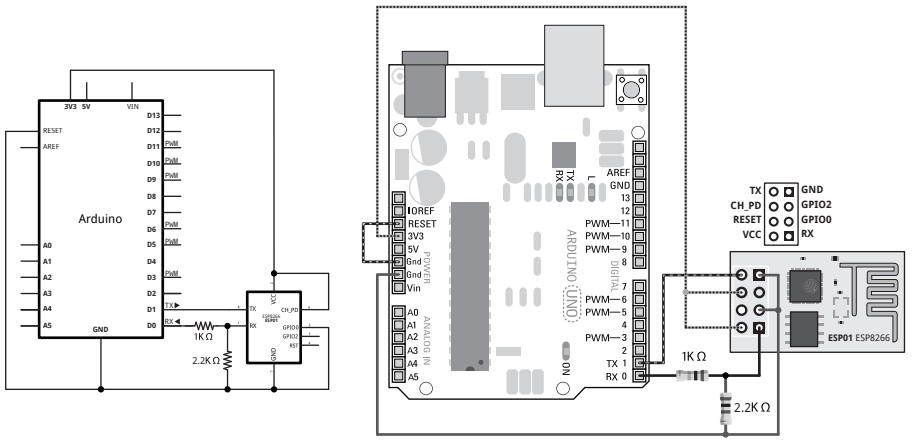
آرایه‌های نمایش داده شده در هر خط از برنامه، از بالا به پایین بیان‌گر اعداد ۰ تا ۹ می‌باشند. در مرحله ی بعد، در بخش ()setup برنامه، با استفاده از تابع ()pinMode درگاه‌های دیجیتال شماره ۰ تا ۶ بورد آردوینو، در وضعیت خروجی قرار می‌گیرند. در بخش ()loop برنامه، در دو مرحله اقدام به تولید اعداد ترتیبی می‌کنیم: مرحله‌ی اول تولید اعداد افزایشی، و مرحله‌ی دوم تولید اعداد کاهشی. برای تولید اعداد در هریک از این مراحل، دو بار از دستور ()for استفاده می‌کنیم. ابتدا اعداد ۰ تا ۹ بر روی سون‌سگمنت نمایش داده می‌شوند. فاصله‌ی زمانی نمایش اعداد، ۱۰۰۰ میلی‌ثانیه است که توسط تابع ()delay ایجاد می‌شود. سپس اعداد ۹ تا ۰ با فاصله‌ی زمانی ۲۵۰ میلی‌ثانیه نمایش داده می‌شوند.

پروژه‌ی نهم: حسگر فاصله‌یاب فراصوتی

در این پروژه می‌خواهیم از طریق دور و نزدیک کردن اشیاء، حسگر فاصله‌یاب فراصوتی (حسگر اولتراسونیک) را فعال نماییم. قطعات مورد نیاز برای انجام پروژه عبارت‌اند از:

بردبورد
سیم‌جامپر
مقاومت ۱ کیلواهم (قهوه ای، مشکی، قرمز)
ال‌ای‌دی
حسگر اولتراسونیک (اس‌آراف۰۵)
بورد آردوینو

همان‌طور که پیش‌تر در فصل ماژول‌ها و شیلدهای آردوینو گفتیم، حسگر فاصله‌یاب فراصوتی یا اولتراسونیک، دارای یک فرستنده و یک گیرنده‌ی امواج مافوق صوت است. امواج ابتدا به سمت مانع فرستاده می‌شوند و پس از برخورد به آن، بازمی‌گردند. برمبنای زمان پژواک موج، می‌توان فاصله‌ی حسگر تا مانع را محاسبه کرد. حسگرهای اولتراسونیک تحت تاثیر رنگ و نور اشیاء مختلف گمراه نمی‌شوند ولی در برابر اجسام متخلخل از کارایی بالایی برخوردار نیستند. در این



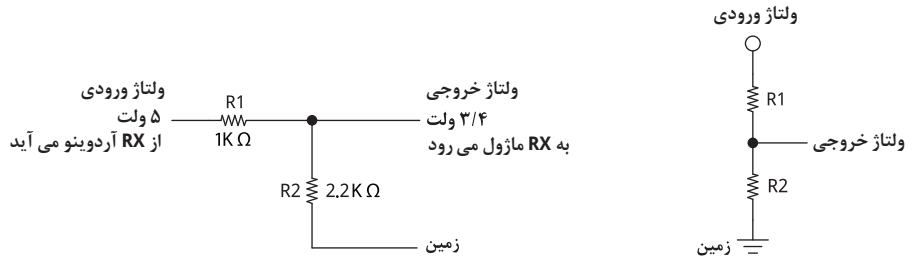
شکل ۶.۲۶: نحوه‌ی اتصال ماژول اینترنت به بورد آردوینو (برای حالت برنامه‌دهی)

ابتـدا در بـورد آردوینـو، درگاه‌هـای RESET و GND را بـه هـم وصـل می‌کنیـم. ایـن کار باعـث می‌شـود کـه بـورد آردوینـو در حالـت ریسـت دایـم قرارگیـرد و میکروکنترلـر آن از دسـترس خـارج شـود. بنابرایـن می‌تـوان از بـورد آردوینـو، به‌عنـوان یـک واسـط اتصـال بیـن مـاژول وای‌فـای و نرم‌افـزار آردوینـو اسـتفاده کـرد و بـه برنامه‌دهـی بـه مـاژول پرداخـت. به‌جـای اتصـال درگاه‌هـای RESET و GND بـورد آردوینـو بـه یکدیگـر، می‌توان میکروکنترلـر موجـود بـر روی بورد آردوینو را از جای خـود در آورد و موقتاً از بورد جدا نمود.

- درگاه VCC ماژول را به درگاه قدرت ۳/۳ ولت آردوینو وصل می‌کنیم.
- درگاه CH_PD را را به ولتاژ ۳/۳ وصل می‌کنیم تا چیپ روشن بماند.
- درگاه GPIO0 بایـد بـه GND متصـل شـود تـا «حالـت برنامه‌دهی به ماژول» فعال شـود.
- درگاه GND ماژول به درگاه GND منبع تغذیه وصل می‌شود.
- درگاه RX مـاژول بـه درگاه RX بـورد آردوینـو (درگاه شـماره ۰ دیجیتال) وصل می‌شـود.

ایـن اتصـال، گرفتـن داده‌هـا توسـط مـاژول را امکان‌پذیـر می‌سـازد. ولتـاژ ارسـال داده‌هـا از بـورد آردوینـو، حـدود ۵ ولـت اسـت. بـا توجـه بـه آن‌کـه مـاژول حاضـر، نمی‌توانـد ولتـاژ ۵ ولـت موجـود در RX آردوینـو را تحمـل نمایـد بنابرایـن مطابـق بـا شـکل‌های ۶.۲۶ و ۶.۲۷ می‌توانیـم دو مقاومـت ۱ کیلواهـم و ۲/۲ کیلواهـم را در مـدار قـرار دهیـم تـا بـه مـاژول آسـیب نرسـد. همان‌طـور کـه قبـلاً گفتـه شـد، بـه مقاومت‌هایـی کـه بـه

این صورت به هم متصل شده باشند اصطلاحاً مقاومت‌های تقسیم‌کننده‌ی ولتاژ می‌گویند که در شکل ۶.۲۷ نمایش داده شده است. این روش اتصال مقاومت‌ها، یکی از ساده‌ترین و ارزان‌ترین راه‌ها برای کاهش ولتاژ است.



شکل ۶.۲۷: مقاومت‌های تقسیم‌کننده‌ی ولتاژ برای حفاظت از ماژول وای‌فای

رابطه‌ی بین ولتاژ ورودی V_{IN}، ولتاژ خروجی V_{OUT}، و دو مقاومت R۱ و R۲ به قرار زیر است:

$$V_{OUT} = V_{IN} \times (R۲ / (R۱ + R۲))$$

مطابق با فرمول بالا، ولتاژ VIN به میزان ۵ ولت از RX آردوینو خارج شده و پس از عبور از مقاومت‌های تقسیم‌کننده‌ی ولتاژ، در قالب VOUT به میزان ۳/۴ ولت تقلیل یافته و به RX ماژول وای‌فای وارد می‌شود.

$$V_{OUT} = ۵ \times (۲۲۰۰ / (۱۰۰۰ + ۲۲۰۰)) = ۵ \times (۲۲۰۰/۳۲۰۰) = ۵ \times ۰/۶۹ = ۳/۴۵ \text{ ولت}$$

ولتاژ ۳/۴۵ ولت، تا حد زیادی به مقدار مطلوب ما که ۳/۳ ولت است، نزدیک بوده و قابل قبول است. در صورتی‌که در جعبه‌ابزار خود مقاومت ۲/۲ کیلواهم ندارید می‌توانید با سری کردن دو مقاومت ۱ کیلواهم و دو مقاومت ۱۰۰ اهم آن را فراهم نمایید.

• درگاه TX ماژول به درگاه TX بورد آردوینو (درگاه شماره ۱ دیجیتال) وصل می‌شود. این اتصال، فرستادن داده‌ها توسط ماژول را امکان‌پذیر می‌سازد. ولتاژ ارسال داده‌ها از ماژول، حدود ۳/۳ ولت است. بنابراین نیازی به نصب مقاومت در مسیر TX نمی‌باشد.

می دانیم که در حالت عادی، برای تبادل داده‌ها بین بورد آردوینو و ماژول باید درگاه RX بورد آردوینو به درگاه TX ماژول و درگاه TX بورد آردوینو به درگاه RX ماژول وصل شوند ولی در این پروژه ما نمی‌خواهیم با بورد آردوینو تبادل داده کنیم بلکه می‌خواهیم از طریق آن، ارتباط ماژول را با نرم‌افزار آردوینو، که

بـر روی رایانـه نصـب شـده اسـت، برقـرار کنیـم. توجـه نماییـد کـه مـاژول نمی‌توانـد مسـتقیما از طریـق درگاه یواس‌بـی بـه رایانـه وصـل شـود، چـون ارتبـاط آن، فقـط از نـوع سـریال و از طریـق درگاه‌هـای RX و TX اسـت. بنابرایـن از بـورد آردوینـو به‌عنـوان مبـدل سـریال بـه یواس‌بـی اسـتفاده می‌کنیـم. یعنـی مـا می‌خواهیـم به‌جـای تبـادل داده بـا میکروکنترلـر ATmega328 موجـود بـر روی بـورد آردوینـو، از میکروپردازشـگر مبـدّل سـریال بـه یواس‌بـی ATmega16U2 موجـود بـر روی بـورد آردوینـو، به‌عنـوان واسـطه‌ی ارتبـاط بـا رایانـه و نرم‌افـزار آردوینـو اسـتفاده کنیـم. بـه همیـن دلیـل، ابتـدا درگاه‌هـای RESET و GND بـورد آردوینـو را بـه هـم وصـل کرده‌ایـم تـا میکروکنترلـر آردوینـو از دسـترس خـارج شـود و سـپس از طریـق درگاه‌هـای RX و TX بـورد آردوینـو، بـرای ارتبـاط بـا رایانـه بهـره گرفتـه ایـم. بنابرایـن بایـد درگاه RX بـورد آردوینـو بـه درگاه RX مـاژول و درگاه TX بـورد آردوینـو بـه درگاه TX مـاژول وصـل شـوند.

ب. حالت معمول اجرای برنامه

پـس از انجـام عملیـات برنامه‌دهـی بـه مـاژول، آن‌را از بـورد آردوینـو جـدا می‌کنیـم. بـرای شـروع بـه‌کار مسـتقل مـاژول، به‌صـورت زیـر عمـل می‌کنیـم:

- درگاه VCC ماژول را به قطب مثبت منبع تغذیه ۳/۳ ولت وصل می‌کنیم.
- درگاه CH_PD را را به ولتاژ ۳/۳ وصل می‌کنیم تا چیپ روشن بماند.
- درگاه GND ماژول به درگاه GND منبع تغذیه وصل می‌شود.

سـایر درگاه‌هـا، متناسـب بـا نیـاز، می‌تواننـد مـورد اسـتفاده قرارگیرنـد. مثـلاً ممکـن اسـت از درگاه GPIO2 مـاژول به‌عنـوان خروجـی سـیگنال بـرای روشـن و خاموش‌کـردن یـک ال‌ای‌دی اسـتفاده شـود.

تنظیم نرم‌افزار آردوینو برای ارتباط با ماژول

در ایـن مرحلـه می‌خواهیـم نرم‌افـزار آردوینـو را بـرای ارتبـاط بـا مـاژول وای‌فـای تنظیـم کنیـم. ابتـدا بایـد وارد نرم‌افـزار آردوینـو شـویم و مـاژول (ESP8266) ESP-01 را به‌عنـوان یـک سـخت‌افزار جدیـد بـه آن معرفـی کنیـم. به‌صـورت پیش‌فـرض، مشـخصات ایـن بـورد بـرای نرم‌افـزار آردوینـو تعریـف نشـده اسـت. بنابرایـن بایـد خودمـان به‌صـورت

```
turn the LED on pin 2 ON<br>");
 client.println("Click <a href=\"/LED=OFF\">here</a>
turn the LED on pin 2 OFF<br>");
 client.println("</html>");
 delay(1);
 Serial.println("Client disonnected");
 Serial.println("");
}
```

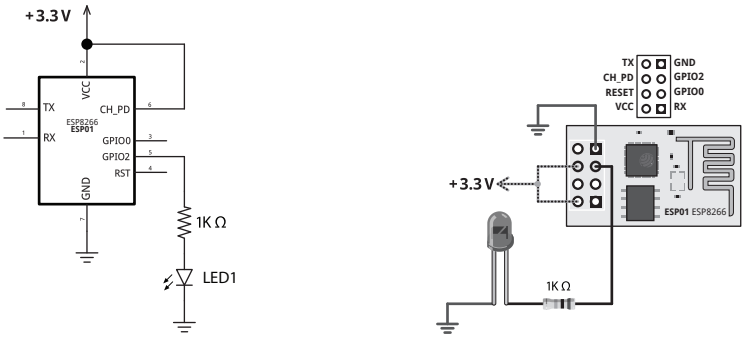
پس از نوشتن اسکچ فوق در نرم‌افزار آردوینو، به مسیر زیر بروید تا بر روی ماژول، بارگذاری شود:

Sketch -> Upload

سپس از طریق مسیر زیر، نمایشگر سریال را باز کنید:

Tools > Serial Monitor

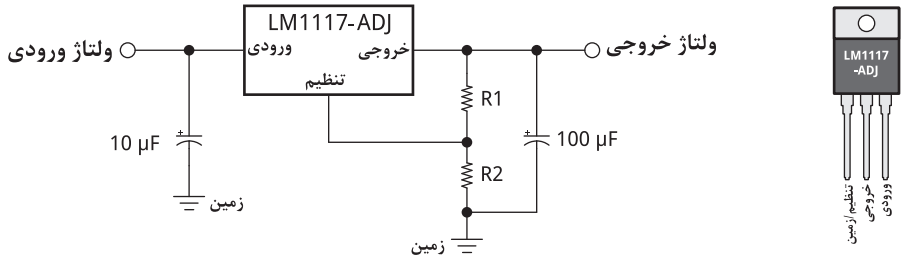
نرخ تبادل داده در نمایشگر سریال را بر روی ۱۱۵۲۰۰ تنظیم کنید. یک آدرس URL (مثلاً http://192.168.1.111) بر روی نمایشگر سریال نمایش داده می‌شود. این آدرس را در صفحه‌ی مرورگر اینترنت خود (مثلاً کروم) وارد کنید. صفحه‌ای باز می‌شود که در آن صفحه می‌توانید از طریق مودم اینترنت، یک ال‌ای‌دی را که به ماژول وای‌فای متصل کرده اید، خاموش و روشن نمایید. نحوه‌ی اتصال ال‌ای‌دی به ماژول وای‌فای در شکل ۶.۲۸ آمده است. همان‌طور که مشاهده می‌کنید، ماژول وای‌فای را در حالت معمول اجرای برنامه (حالت ب) قرار داده‌ایم و ال‌ای‌دی را به درگاه دیجیتال شماره ۲ ماژول وای‌فای (درگاه GPIO2) متصل نموده ایم. در این مدار می‌توانید به‌جای ال‌ای‌دی، از یک رله استفاده کنید تا بتوانید تجهیزات مختلف الکتریکی را کنترل نمایید. در این صورت رعایت کلیه‌ی نکات ایمنی برای جلوگیری از خطر برق‌گرفتگی الزامی است.



شکل ۶.۲۸: نحوه‌ی اتصال ال‌ای‌دی به ماژول اینترنت

مدار تثبیت‌کننده‌ی ولتاژ برای ماژول

پیش‌تر گفتیم که محدوده‌ی ولتاژ قابل قبول برای تغذیه‌ی ماژول ارتباط با اینترنت، و یا ماژول وای‌فای، بین ۲/۵ تا ۳/۶ ولت است و به‌طور معمول می‌توان از ولتاژ ۳/۳ ولت با شدت جریان ۲۵۰ میلی‌آمپر استفاده کرد. در دسترس‌ترین منبع تغذیه با این مشخصات، درگاه قدرت ۳/۳ ولت بورد آردوینو است. مشکلی که وجود دارد آن است که در بسیاری از بوردهای آردوینو، جریان الکتریکی پایدار این درگاه، کمتر از ۲۵۰ میلی‌آمپر است که عملکرد ماژول را با اختلال مواجه می‌کند. در صورت مواجهه با این مشکل می‌توانید از درگاه قدرت ۵ ولت بورد آردوینو و یا یک منبع تغذیه‌ی خارجی، مثلاً یک آداپتور، استفاده کنید ولی به این نکته توجه داشته باشید که چون ماژول وای‌فای، نسبت به ولتاژهای بالا آسیب‌پذیر است، حتماً باید سطح ولتاژ را توسط یک تثبیت‌کننده‌ی ولتاژ، کاهش داد. در شکل ۶.۲۹ یک مدار کاهش ولتاژ نمایش داده شده است که بر مبنای یک تثبیت‌کننده‌ی ولتاژ به نام ADJ-LM1117 کار می‌کند.



شکل ۶.۲۹: مدار تثبیت‌کننده‌ی ولتاژ

همان‌طور که در شکل ۶.۲۹ می‌بینید، با اضافه‌کردن دو عدد خازن الکترولیت و دو عدد مقاومت، به ADJ-LM1117 می‌توان یک مدار تثبیت‌کننده‌ی ولتاژ تشکیل داد و ولتاژ خروجی مدار را بر روی مقدار دلخواه تنظیم کرد. در مدار فوق، ولتاژ ورودی حداکثر ۱۵ ولت (مستقیم) است ولی ولتاژ خروجی آن، مطابق با فرمول زیر، بین ۱/۲۵ تا ۱۳/۸ ولت (مستقیم) متغیر است:

$$V_{\text{خروجی}} = ۱/۲۵ \left(۱ + \frac{R۲}{R۱} \right)$$

پروژه‌ی دوازدهم: فشرده‌سازی آردوینو در یک آی‌سی

در برخی از پروژه‌های آردوینو، به حجم اندکی از حافظه و تعداد محدودی از درگاه‌های ورودی و خروجی نیاز داریم. از سوی دیگر ممکن است به تعداد زیادی میکروکنترلر با هزینه‌ی اندک نیازمند باشیم. در این‌گونه موارد می‌توانیم از میکروکنترلرهایی مانند ATtiny13 استفاده کنیم. مثلاً برای تهیه‌ی یک کارت تبریک موزیکال، می‌توانیم عملکرد کلید و بلندگو را توسط این میکروکنترلر کنترل‌کنیم و موسیقی را در حافظه‌ی آن ذخیره نماییم. در این پروژه می‌خواهیم روشن و خاموش‌شدن یک ال‌ای‌دی را با استفاده از آی‌سی ATtiny13 کنترل‌کنیم. برنامه‌دهی به این آی‌سی توسط آردوینو انجام می‌شود. سپس و سپس این آی‌سی از آردوینو جدا شده و به‌صورت مستقل به کار خود ادامه خواهد داد.
قطعات مورد نیاز برای انجام این پروژه عبارت‌اند از:

بردبورد
سیم‌جامپر
مقاومت ۱ کیلواهم (قهوه ای، مشکی، قرمز)
ال‌ای‌دی
آی‌سی (اَت‌تاینی۱۳)
بورد آردوینو

میکروکنترلر ATtiny13 یک آی‌سی خام است و دارای هشت پایه به شرح زیر می‌باشد.

- پایه ۱: درگاه دیجیتال ۵، درگاه آنالوگ ۰ و درگاه راه‌اندازی مجدد
- پایه ۲: درگاه دیجیتال ۳ و درگاه آنالوگ۳
- پایه ۳: درگاه دیجیتال ۴ و درگاه آنالوگ۲
- پایه ۴: قطب منفی (اتصال زمین)
- پایه ۵: درگاه دیجیتال ۰ و درگاه خروجی سریال
- پایه ۶: درگاه دیجیتال ۱ و درگاه ورودی سریال
- پایه ۷: درگاه دیجیتال ۲، درگاه آنالوگ ۱، و ورودی ساعت سریال
- پایه ۸: درگاه تغذیه ۲/۷ تا ۵/۵ ولت

در شکل ۶.۳۱ ترتیب قرارگیری این پایه‌ها آمده است.

همان‌طور که می‌بینید، برای بارگذاری اسکچ بر روی آی‌سی ATtiny13 از آردوینو به‌عنوان پروگرامر استفاده می‌کنیم، چراکه نرم‌افزار آردوینو، به‌طور مستقیم نمی‌تواند با ATtiny13 ارتباط برقرار کند بلکه باید از طریق بورد آردوینو به‌عنوان واسط، عملیات برنامه‌دهی به ATtiny13 را انجام دهد.

اکنون ال‌ای‌دی شروع به چشمک زدن می‌کند. همان‌طور که می‌بینید، مدت زمان روشن و خاموش ماندن آن مطابق با اسکچ بارگذاری شده بر روی ATtiny13 است. در این مرحله می‌توان ATtiny13 را از آردوینو جدا نمود و به‌صورت مستقل از آن استفاده کرد. برای تغذیه‌ی مدار مستقل، می‌توان از باتری و یا سایر منابع تغذیه با دامنه‌ی ۲/۷ تا ۵/۵ ولت استفاده نمود.

لطفاً توجه نمایید که علاوه بر ATtiny13، آی‌سی‌های قابل برنامه‌ریزی دیگری نیز وجود دارند (مانند ATtiny84 و ATtiny85) که برای راه‌اندازی هریک از آن‌ها باید مشابه با همین پروژه، فایل مختص به آن نمونه را به‌صورت دستی در بخش Boards manager به نرم‌افزار آردوینو اضافه کرد و تنظیمات مربوطه را بر اساس ویژگی آن نمونه‌ها انجام داد.