

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

اَللّٰهُمَّ صَلِّ عَلٰى مُحَمَّدٍ وَّآلِ مُحَمَّدٍ وَّعَجِّلْ فَرَجَهُمْ



ساخت مدارهای دیجیتال و میکروکنترلر

رشته الکترونیک

گروه برق و رایانه

شاخه فنی و حرفه‌ای

پایه یازدهم دوره دوم متوسطه





وزارت آموزش و پرورش
سازمان پژوهش و برنامه‌ریزی آموزشی



- نام کتاب: ساخت مدارهای دیجیتال و میکروکنترلر - ۲۱۱۲۷۸
- پدیدآورنده: سازمان پژوهش و برنامه‌ریزی آموزشی
- مدیریت برنامه‌ریزی درسی و تألیف: دفتر تألیف کتاب‌های درسی فنی و حرفه‌ای و کاردانش
- شناسه افزوده برنامه‌ریزی و تألیف: شهرام نصیری سوادکوهی، شهرام خدادادی، محمود شبانی و احمد توکلی (اعضای شورای برنامه‌ریزی)
- محمود شبانی، مجید مکرم نجف‌آباد، اکبر تیمور پارسایی، مرتضی عزیزی، داود افراسیاب و شهرام خدادادی (اعضای گروه تألیف) - محمود شبانی و شهرام نصیری سوادکوهی (ویراستار فنی و ادبی)
- مدیریت آماده‌سازی هنری: اداره کل نظارت بر نشر و توزیع مواد آموزشی
- شناسه افزوده آماده‌سازی: احمدرضا امینی (مدیر امور فنی و چاپ) - جواد صفری (مدیر هنری) - پارمیس جوشقانی (صفحه‌آرا) - مریم کیوان (طراح جلد) - زهرا رشیدی‌مقدم، سیف‌الله بیک‌محمدولیوند، فریبا سیر، سیما لطفی (امور آماده‌سازی)
- نشانی سازمان: تهران: خیابان ایرانشهر شمالی ساختمان شماره ۴ آموزش و پرورش (شهید موسوی)
- تلفن: ۸۸۸۳۱۱۶۱-۹، دورنگار: ۸۸۳۰۹۲۶۶، کد پستی: ۱۵۸۴۷۴۷۳۵۹
- وب‌گاه: www.chap.sch.ir و www.irtextbook.ir
- ناشر: شرکت چاپ و نشر کتاب‌های درسی ایران: تهران-کیلومتر ۱۷ جاده مخصوص کرج - خیابان ۶۱ (داروپخش)
- تلفن: ۴۴۹۸۵۱۶۱-۵، دورنگار: ۴۴۹۸۵۱۶۰ صندوق پستی: ۳۷۵۱۵-۱۳۹
- چاپخانه: شرکت چاپ و نشر کتاب‌های درسی ایران «سهامی خاص»
- سال انتشار و نوبت چاپ: چاپ اول ۱۴۰۵

کلیه حقوق مادی و معنوی این کتاب متعلق به سازمان پژوهش و برنامه‌ریزی آموزشی وزارت آموزش و پرورش است و هرگونه استفاده از کتاب و اجزای آن به صورت چاپی و الکترونیکی و ارائه در پایگاه‌های مجازی، نمایش، اقتباس، تلخیص، تبدیل، ترجمه، عکس‌برداری، نقاشی، تهیه فیلم و تکثیر به هر شکل و نوع بدون کسب مجوز از این سازمان ممنوع است و متخلفان تحت پیگرد قانونی قرار می‌گیرند.



ملت شریف ما اگر در این انقلاب بخواهد پیروز شود باید دست از آستین
برآرد و به کار بپردازد. از متن دانشگاه‌ها تا بازارها و کارخانه‌ها و مزارع و
باغستان‌ها تا آنجا که خودکفا شود و روی پای خود بایستد.
امام خمینی (قَدَسَ سِرُّهُ)

این کتاب در سال‌های ۱۴۰۲ و ۱۴۰۳ از منظر چهار گروه از دنیای کار (کارفرمایان بزرگ، خبرگان حرفه‌ای، فروشندگان خدمات و کالا و انجمن‌های حرفه‌ای) و چهار گروه از دنیای آموزش (هنرآموزان خبره، مربیان سازمان آموزش فنی و حرفه‌ای، هنرجویان پایه دوازدهم و فارغ‌التحصیلان شاغل) مورد ارزشیابی کشوری قرار گرفت و نتایج آن در محتوا اعمال گردید.

۱.....	ساخت برد SMD	پودمان اول
۲.....	واحد یادگیری ۱: دمونتاژ SMD	
۲۷.....	واحد یادگیری ۲: مونتاژ SMD	
۴۷.....	ساخت مدارهای دیجیتالی پایه	پودمان دوم
۴۸.....	واحد یادگیری ۱: اجرای مدار عملگرهای منطقی	
۷۹.....	واحد یادگیری ۲: اجرای مدارهای منطقی	
۱۰۳.....	پیاده‌سازی مدارهای کاربردی دیجیتال	پودمان سوم
۱۰۴.....	واحد یادگیری ۱: پیاده‌سازی مدارهای ترکیبی	
۱۳۴.....	واحد یادگیری ۲: پیاده‌سازی مدارهای ترتیبی	
۱۸۳.....	انتخاب میکروکنترلر (Micro Controller) و برنامه‌ریزی آن	پودمان چهارم
۱۸۴.....	واحد یادگیری ۱: انتخاب میکروکنترلر	
۲۰۸.....	واحد یادگیری ۲: برنامه نویسی میکروکنترلر	
۲۴۹.....	پیاده‌سازی مدارهای کاربردی با میکروکنترلر	پودمان پنجم
۲۵۰.....	واحد یادگیری ۱: ورودی‌ها و خروجی‌های دیجیتال در میکروکنترلر ESP32	
۲۷۷.....	واحد یادگیری ۲: ورودی و خروجی آنالوگ و کنترل موتور با ESP32	
۳۰۰.....	منابع	

شرایط در حال تغییر دنیای کار در مشاغل گوناگون، توسعه فناوری‌ها و تحقق توسعه پایدار، ما را بر آن داشت تا برنامه‌های درسی و محتوای کتاب‌های درسی را در ادامه تغییرات پایه‌های قبلی براساس نیاز کشور و مطابق با رویکرد سند تحول بنیادین آموزش و پرورش و برنامه درسی ملی جمهوری اسلامی ایران در نظام جدید آموزشی بازطراحی و تألیف کنیم. مهم‌ترین تغییر در کتاب‌ها، آموزش و ارزشیابی مبتنی بر شایستگی است. شایستگی، توانایی انجام کار واقعی به‌طور استاندارد و درست تعریف شده است. توانایی شامل دانش، مهارت و نگرش می‌شود. در رشته تحصیلی حرفه‌ای شما، چهار دسته شایستگی در نظر گرفته شده است:

- ۱ شایستگی‌های فنی برای جذب در بازار کار مانند تسلط بر ساخت و پیاده‌سازی مدارهای دیجیتال و سامانه‌های مبتنی بر میکروکنترلر
- ۲ شایستگی‌های غیرفنی برای پیشرفت و موفقیت در آینده مانند نوآوری و مصرف بهینه
- ۳ شایستگی‌های فناوری اطلاعات و ارتباطات مانند کار با نرم‌افزارها
- ۴ شایستگی‌های مربوط به یادگیری مادام‌العمر مانند کسب اطلاعات از منابع دیگر

بر این اساس دفتر تألیف کتاب‌های درسی فنی و حرفه‌ای و کاردانش مبتنی بر اسناد بالادستی و با مشارکت متخصصان برنامه‌ریزی درسی فنی و حرفه‌ای و خبرگان دنیای کار مجموعه اسناد برنامه درسی رشته‌های شاخه فنی و حرفه‌ای را تدوین نموده‌اند که مرجع اصلی و راهنمای تألیف کتاب‌های درسی هر رشته است.

این درس، دروس پایه شایستگی‌های فنی و کارگاهی است که ویژه رشته الکترونیک تألیف شده است. کسب شایستگی‌های این کتاب برای موفقیت آینده شغلی و حرفه‌ای شما بسیار ضروری است. هنرجویان عزیز سعی نمایید؛ تمام شایستگی‌های آموزش داده شده در این کتاب را کسب و در فرایند ارزشیابی به اثبات رسانید.

کتاب درسی ساخت مدارهای دیجیتال و میکروکنترلر شامل پنج پودمان است و هر پودمان دارای دو واحد یادگیری است و هر واحد یادگیری از چند مرحله کاری تشکیل شده است. شما هنرجویان عزیز پس از یادگیری هر پودمان می‌توانید شایستگی‌های مربوط به آن را کسب نمایید. شرط لازم برای کسب شایستگی در هر پودمان، کسب شایستگی در هر دو واحد یادگیری آن پودمان است. در صورت احراز نشدن شایستگی در هر یک از واحدهای یادگیری پس ارزشیابی اول، فرصت جبران و ارزشیابی دوباره تا پایان سال تحصیلی وجود دارد. ارزشیابی هر واحد یادگیری نیز از دو بخش نمره مستمر و نمره شایستگی فنی تشکیل شده است و شرط لازم برای محاسبه این دو بخش، کسب شایستگی در بخش شایستگی‌های غیرفنی است.

همچنین علاوه بر کتاب درسی شما امکان استفاده از سایر اجزای بسته آموزشی که برای شما طراحی و تألیف شده است، وجود دارد. یکی از این اجزای بسته آموزشی کتاب همراه هنرجو می‌باشد که برای انجام فعالیت‌های موجود در کتاب درسی باید استفاده نمایید. کتاب همراه خود را می‌توانید هنگام آزمون و فرایند ارزشیابی نیز همراه داشته باشید. سایر اجزای بسته آموزشی دیگری نیز برای شما در نظر گرفته شده است که با مراجعه به وبگاه رشته خود با نشانی www.oerp.ir می‌توانید از عناوین آن مطلع شوید.

فعالیت‌های یادگیری در ارتباط با شایستگی‌های غیرفنی از جمله مدیریت منابع، اخلاق حرفه‌ای، حفاظت از محیط زیست و شایستگی‌های یادگیری مادام‌العمر و فناوری اطلاعات و ارتباطات همراه با شایستگی‌های فنی طراحی و در کتاب درسی و بسته آموزشی ارائه شده است. شما هنرجویان عزیز کوشش نمایید این شایستگی‌ها را در کنار شایستگی‌های فنی آموزش ببینید، تجربه کنید و آنها را در انجام فعالیت‌های یادگیری به کار گیرید.

رعایت نکات ایمنی، بهداشتی و حفاظتی از اصول انجام کار است؛ لذا توصیه‌های هنرآموز محترمتان در خصوص رعایت مواردی که در کتاب آمده است، در انجام کارها جدی بگیرید.

امیدواریم با تلاش و کوشش شما هنرجویان عزیز و هدایت هنرآموزان گرامی، گام‌های مؤثری در جهت سربلندی و استقلال کشور و پیشرفت اجتماعی و اقتصادی و تربیت مؤثر و شایسته جوانان برومند میهن اسلامی برداشته شود.

سخنی با هنرآموزان گرامی

در راستای تحقق اهداف سند تحول بنیادین آموزش و پرورش و برنامه درسی ملی جمهوری اسلامی ایران و نیازهای متغیر دنیای کار و مشاغل، برنامه درسی رشته الکترونیک طراحی و براساس آن محتوای آموزشی نیز تألیف گردید. کتاب حاضر از مجموعه کتاب‌های کارگاهی می‌باشد که برای سال یازدهم تدوین و تألیف گردیده است. این کتاب دارای پنج پودمان است و شرط لازم برای کسب شایستگی در هر پودمان کسب شایستگی در هر دو واحد یادگیری آن پودمان است. در صورت احراز نشدن شایستگی در هریک از واحدهای یادگیری پس ارزشیابی اول، فرصت جبران و ارزشیابی دوباره تا پایان سال تحصیلی وجود دارد. ارزشیابی هر واحد یادگیری نیز از دو بخش نمره مستمر و نمره شایستگی فنی تشکیل شده است و شرط لازم برای محاسبه این دو بخش، کسب شایستگی در بخش شایستگی‌های غیرفنی است. از ویژگی‌های دیگر این کتاب طراحی فعالیت‌های یادگیری ساخت یافته در ارتباط با شایستگی‌های فنی و غیرفنی از جمله مدیریت منابع، اخلاق حرفه‌ای و مباحث زیست محیطی است. این کتاب جزئی از بسته آموزشی تدارک دیده شده برای هنرجویان است که لازم است از سایر اجزای بسته آموزشی مانند کتاب همراه هنرجو، نرم افزار و فیلم آموزشی در فرایند یادگیری استفاده شود. کتاب همراه هنرجو در هنگام یادگیری، ارزشیابی و انجام کار واقعی مورد استفاده قرار می‌گیرد. شما می‌توانید برای آشنایی بیشتر با اجزای بسته یادگیری، روش‌های تدریس کتاب، شیوه ارزشیابی مبتنی بر شایستگی، مشکلات رایج در یادگیری محتوای کتاب، بودجه‌بندی زمانی، نکات آموزشی شایستگی‌های غیرفنی، آموزش ایمنی و بهداشت و دریافت راهنما و پاسخ فعالیت‌های یادگیری و تمرین‌ها به کتاب راهنمای هنرآموز این درس مراجعه کنید. رعایت ایمنی و بهداشت، شایستگی‌های غیرفنی و مراحل کلیدی براساس استاندارد از ملزومات کسب شایستگی می‌باشند.

کتاب شامل پودمان‌های ذیل است:

پودمان اول: ساخت برد SMD: در این پودمان، هنرجویان با مفاهیم پایه و اصول جداسازی قطعات و روش‌های استاندارد جداسازی و نصب قطعات SMD بر روی بردهای مدار چاپی آشنا می‌شوند و مهارت‌های لازم برای ساخت برد با قطعات SMD را کسب می‌کنند.

پودمان دوم: ساخت مدارهای دیجیتالی پایه: این پودمان بر پایه مفاهیم سامانه‌های دیجیتال بنا شده است؛ در این بخش، مباحثی همچون تبدیل سامانه‌های اعداد، سطوح منطقی، توابع منطقی، ساده‌سازی آنها و در نهایت پیاده‌سازی توابع منطقی با استفاده از دروازه‌های منطقی مورد آموزش قرار می‌گیرد.

پودمان سوم: پیاده‌سازی مدارهای کاربردی دیجیتال: در این پودمان، مدارهای ترکیبی و ترتیبی متداول مورد تحلیل و تشریح قرار گرفته و با هدف تقویت مهارت‌های کاربردی، پروژه‌های پیشنهادی در زمینه طراحی و اجرای مدارهای دیجیتال انجام می‌پذیرد.

پودمان چهارم: انتخاب میکروکنترلر: تمرکز این پودمان بر فرایند انتخاب میکروکنترلر متناسب با نیاز پروژه است. همچنین، آموزش مبانی زبان برنامه‌نویسی و کار با محیط‌های توسعه (IDE) جهت نوشتن برنامه‌های ساده برای میکروکنترلر انتخاب شده در این بخش ارائه می‌گردد.

پودمان پنجم: پیاده‌سازی مدارهای کاربردی با میکروکنترلر: در این پودمان، پس از آشنایی با پیکربندی پایه‌های ورودی/خروجی میکروکنترلر، هنرجویان به طراحی و اجرای پروژه‌های کنترل هوشمند می‌پردازند. این فرایند شامل برنامه‌نویسی به زبان C، عملیات برنامه‌ریزی میکروکنترلر و پیاده‌سازی مدار عملی پروژه خواهد بود.

امید است که با تلاش و کوشش شما همکاران گرامی اهداف پیش‌بینی شده برای این درس محقق گردد.

دفتر برنامه‌ریزی و تألیف کتاب‌های درسی فنی و حرفه‌ای و کار دانش



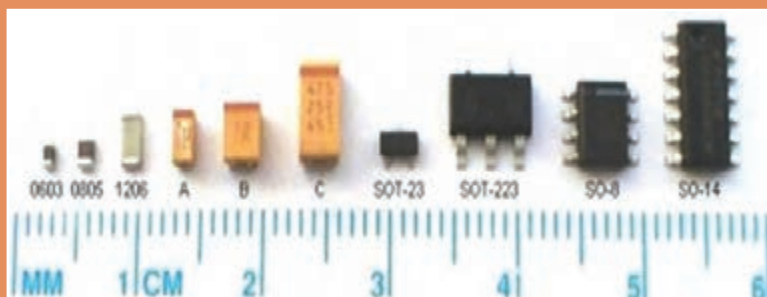
نظر سنجی کتاب درسی





پودمان اول

ساخت برد SMD



آیا تاکنون نسبت به ساخت برد مدار چاپی برای مداری که دارای ICهای متعدد است اقدام کرده‌اید؟ آیا با کار طاقت‌فرسای موردنیاز برای سوراخ کردن محل پایه‌های چندین IC آشنایی دارید؟ یادآور می‌شود که نه تنها تعداد محل‌هایی که باید سوراخ شوند زیاد است، بلکه با توجه به نوع قطعه، چیدمان سوراخ‌های ایجاد شده باید به گونه‌ای مرتب باشند که پایه‌های ICهای دوردیفه و چهارطرفه بتوانند به آسانی در آن قرار گیرند. برای حل این مشکلات از قطعات نصب سطحی یا SMD استفاده می‌کنیم. قطعات نصب سطحی نیازی به سوراخ کردن محل پایه‌ها ندارند که خود دلیل دیگری برای محبوبیت فناوری SMD در صنعت الکترونیک است. در این فناوری پایه‌ها فقط در یک سطح قرار می‌گیرند. این امر موجب شده است تا بتوانیم قطعه‌های SMD را در هر دو سطح برد مدار چاپی دو لایه قرار دهیم. این ویژگی عامل دیگری برای کوچک‌تر شدن مدارهای مبتنی بر نصب سطحی نسبت به مدارهای با قطعات پایه‌دار است.

در صنعت الکترونیک قطعات متعددی وجود دارند که با توجه به ویژگی‌های مخصوص خود، به صورت SMD ساخته می‌شوند. یکی از اساسی‌ترین اشکال فناوری SMD این است که نمی‌توانیم آنها را مانند قطعات پایه‌دار بر روی پرده‌برد قرار دهیم. برای کار با این نوع قطعات در آزمایشگاه، یا آزمایش آنها باید برد مدار چاپی خاص طراحی و چاپ نماییم.

واحد یادگیری ۱: دمونتاژ SMD

آیا تا به حال فکر کرده‌اید

- در بردهای تجهیزات الکترونیکی مانند تلفن همراه و لپ تاپ از چه نوع قطعاتی استفاده شده است؟
 - ابعاد قطعات SMD در چه حدودی است؟
 - قطعات الکترونیکی SMD چه مزایایی نسبت به قطعات مجزا (discrete) دارند؟
 - برای مونتاژ قطعات SMD از چه ابزارها و تجهیزاتی استفاده می‌شود؟
 - در مدارهای الکترونیکی کوچک، ابعاد قطعات به کار رفته، چقدر است؟
 - با کوچک تر شدن ابعاد قطعات، چه ابزارهایی برای شناسایی و کار با آنها نیاز است؟
 - برای تولید انبوه بردهای الکترونیکی با قطعات SMD، از چه تجهیزاتی استفاده می‌شود؟
 - مونتاژ قطعات الکترونیکی SMD با مونتاژ قطعات مجزا (discrete) چه تفاوت‌هایی دارد؟
 - چگونه می‌توان به تعمیر بردهای SMT پرداخت؟
 - چگونه می‌توان قطعات SMD را از روی برد جدا کرد تا برد SMT آسیب نبیند؟
 - هنگام پیاده کردن قطعات SMD چه نکات ایمنی باید رعایت شود؟
 - لحیم کاری روی قطعات SMD با لحیم کاری روی قطعات TH چه تفاوت‌هایی دارد؟
- قابلیت نصب سطحی و کوچک بودن ابعاد، دو ویژگی مهم برای قطعات SMD هستند. کوچک بودن قطعه به معنی به حداقل رسیدن ابعاد کل مدار است. امروزه نیاز به کوچک کردن مدار در ساخت بسیاری از وسیله‌های الکترونیکی مانند تلفن همراه، تبلت، لپ تاپ، ماشین حساب و دوربین‌های ویدئویی وجود دارد و عامل مهمی در گسترش فناوری SMD در صنعت ساخت وسایل الکترونیکی به شمار می‌آید. هر چه ابعاد برد مدار چاپی کوچک تر و طول نوارهای مسی روی برد مدار چاپی کوتاه تر باشد، ظرفیت خازنی بین مسیرها کاهش می‌یابد. از طرفی نبود پایه‌های بلند به معنی حذف اندوکتانس القایی و ظرفیت خازنی مزاحم و پراکنده در مدارهای الکترونیکی، مخصوصاً در مدارهای فرکانس بالا است. این دو عامل موجب شده است که قطعات SMD برای استفاده در مدارهای فرکانس بالا مانند مدارهای تلفن همراه، صوتی و تصویری و ویدئو پروژکتور در اولویت قرار داشته باشند.
- در این واحد یادگیری ابتدا قطعات SMD، ابزار و دستگاه‌های آن شرح داده می‌شود، سپس به دمونتاژ یا پیاده‌سازی قطعات SMD می‌پردازیم. پیاده‌سازی قطعات SMD دارای پیچیدگی‌های خاص است که در فرایند اجرای کار باید به آن توجه شود. از آنجا که روی بردهای SMD قطعات متعددی که دارای کارایی‌های متفاوت هستند نصب شده است، ضرورت دارد هنگام دمونتاژ قطعات، به محدودیت‌ها و استانداردهای تعریف شده برای برد و قطعات نصب شده روی آن دقت کنیم و از ابزار مرتبط و ویژه دمونتاژ استفاده نماییم. همچنین در فرایند کار به اجرای نکات ایمنی توجه خاص داشته باشید.

استاندارد عملکرد

انتخاب ابزار، مواد و تجهیزات مناسب و پیاده‌سازی قطعات SMD از روی برد اوراقی براساس استانداردهای تعریف شده

مواد، ابزارها و تجهیزات مورد نیاز: هویه هوای گرم با نازل‌های متفاوت، برد اوراقی، ابزار ایجاد خلأ (وکیوم - vacuum)، پنس، اسپری پاک‌کننده برد، خمیر فلاکس ویژه قطعات SMD، برد اوراقی، دستمال نخی، ماسک، قطعات SMD، برد اوراقی SMD مدار چاپی.

معرفی ابزارهای مورد نیاز برای دمونتاژ و مونتاژ قطعات SMD

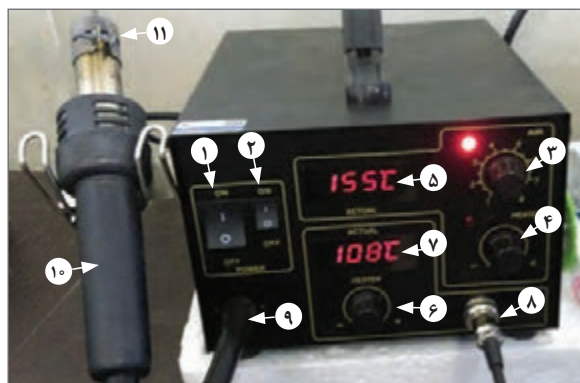


شکل ۱- یک دستگاه هویه هوای گرم

■ هویه هوای گرم مخصوص قطعات SMD:

هویه هوای گرم (Heater) از ابزار و دستگاه‌های SMD است که برای لحیم‌کاری در هنگام نصب (مونتاژ) و پیاده‌سازی (دمونتاژ) قطعات نصب سطحی SMD به کار می‌رود. در شکل ۱ یک دستگاه هویه هوای گرم نشان داده شده است. در یک هویه هوای گرم (هیتر)، هوای گرم به وسیله دستگاه کمپرسور هوا (تولید فشار هوا)، به صورت فشرده تولید می‌شود. این هوا پس از برخورد با رشته (المنت) گرم‌کننده که روی دسته قرار دارد گرم می‌شود و باد گرم از نازل (خروجی) خارج می‌شود. با استفاده از کلید یا ولوم روی دستگاه مرکزی می‌توانیم دما یا میزان هوا را تنظیم کنیم.

برای آشنایی بیشتر با دستگاه هویه هوای گرم، قسمت‌های مختلف یک نمونه دستگاه در شکل ۲ نشان داده شده است. این هویه دارای دو بخش هوای گرم و هویه قلمی است. کار هریک از قسمت‌ها به شرح زیر است:



شکل ۲- قسمت‌های مختلف یک نمونه دستگاه هویه هوای گرم

- ۱ کلید روشن و خاموش شماره ۱: روشن یا خاموش کردن قسمت هویه هوای گرم
- ۲ کلید روشن و خاموش شماره ۲: روشن یا خاموش کردن قسمت هویه قلمی مونتاژ SMD
- ۳ ولوم هوا AIR: تنظیم میزان هوای خروجی از هویه هوای گرم
- ۴ ولوم گرم‌کن شماره ۱ (Heater): تنظیم میزان دمای هوای خروجی هیتر
- ۵ نمایشگر هفت قطعه‌ای شماره ۱ (7Segment): نمایش دمای تنظیم شده برای هویه هوای گرم
- ۶ ولوم گرم‌کن شماره ۲ (Heater): تعیین میزان دمای هویه قلمی
- ۷ نمایشگر هفت قطعه‌ای شماره ۲ (7Segment): نمایش دمای تنظیم شده برای هویه قلمی
- ۸ سوکت شماره ۵: اتصال هویه قلمی SMD به دستگاه هویه هوای گرم
- ۹ خروجی جهت اتصال دسته هویه هوای گرم: اتصال شیلنگ مربوط به هویه هوای گرم
- ۱۰ دسته مخصوص هویه هوای گرم: محل قرار گرفتن المنت و خروج هوای گرم
- ۱۱ نازل: اتصال نازل مناسب با توجه به منطقه کار در مراحل مونتاژ و دمونتاژ



در مورد انواع دیگر هیتر در اینترنت و یا فروشگاه‌ها و تعمیرگاه تحقیق نموده و در کلاس ارائه نمایید.

معمولاً تنظیمات قسمت هویه هوای گرم با روش‌های خاص، مثلاً کشیدن خط جداساز رنگی از تنظیمات هویه قلمی SMD، تفکیک می‌شود.



هنگام استفاده از هویه هوای گرم، با چه خطراتی مواجه هستیم و چه نکات ایمنی درباره سلامت خودمان و سالم ماندن هویه را باید رعایت کنیم. از طریق بارش فکری موضوع خود را مطرح و جمع‌بندی نمایید و به‌صورت یک فهرست وارسی به کلاس ارائه دهید.

■ هویه SMD



هویه‌های SMD انواع مختلفی دارد. این هویه‌ها اکثراً با هیترهای هوای گرم ارائه می‌شود. نوع دیگری هویه SMD به‌صورت پنسی (Clip Forceps) نیز وجود دارد. این هویه‌ها برای برداشتن قطعات دو پایه و موارد خاص دیگر به‌کار می‌رود. شکل ۳ نمونه‌ای از این نوع هویه‌ها را نشان می‌دهد.

این هویه‌ها با توجه به اینکه برای مونتاژ و دیمونتاژ بردهای SMT به‌کار می‌رود نسبت به هویه‌های معمولی نیاز به تولید درجه حرارت بالاتری دارد. هویه‌های SMD انواع مختلف دارد. این هویه‌ها اکثراً با هیترهای هوای گرم ارائه می‌شود. نوک این هویه‌ها نیز نسبت به نوع معمولی نازک‌تر و تیزتر است. در شکل ۴ سه نمونه نوک هویه را مشاهده می‌کنید که برای کاربردهای خاص در SMD به‌کار می‌روند. روش صحیح در دست گرفتن هیتر در شکل ۵ نشان داده شده است.

شکل ۳- هویه مخصوص SMD و هویه پنسی



ب- کاربر راست‌دست



الف- کاربر چپ‌دست



شکل ۴- سه نمونه نوک هویه SMD

شکل ۵- چگونگی صحیح در دست گرفتن هیتر



شکل ۶- چگونگی باز کردن و تعویض نوک هویه

یکی از چالش‌های کار با هویه، تعویض المنت و نوک هویه است. شکل ۶ مراحل و نحوه تعویض نوک هویه‌های هیتری را نشان می‌دهد.

با مراجعه به رسانه‌های مختلف درباره نحوه تعویض المنت هویه‌های هیتری تحقیق کنید و گزارش آن را در کلاس ارائه دهید.

فعالیت



با مراجعه به رسانه‌های مختلف درباره میزان توان و درجه حرارت انواع هویه‌های قلمی و هویه‌های هوای گرم قطعات نصب سطحی تحقیق کنید و نتایج را با توان و میزان حرارت انواع هویه‌های معمولی مقایسه نمایید، سپس گزارش را به صورت جدولی به کارگاه ارائه دهید.

جستجو کنید



مواد مورد نیاز برای لحیم کاری

■ **روغن لحیم یا خمیر فلاکس (Flux):** خمیر فلاکس یک روغن مخصوص برای کار با هیترهای صنعتی است. این روغن مخصوص، فرایند مونتاژ و دهمونتاژ قطعات SMD را سرعت می‌بخشد و آسان می‌کند. عملکرد این خمیر مشابه عملکرد روغن لحیم در لحیم کاری‌های معمولی است. شکل ۷ خمیر فلاکس در دو بسته‌بندی پمپی و جعبه‌ای را نشان می‌دهد.



شکل ۷- انواع خمیر فلاکس

ماده فلاکس یا روغن لحیم SMD به صورت مایع نیز وجود دارد. این ماده در موارد خاص به کار می‌رود. برای هریک از انواع فلاکس‌ها، دستورالعمل کاربرد در ارتباط با میزان مصرف، نوع کار و محل استفاده تنظیم شده است.



توجه





مواد مورد استفاده در لحیم کاری شیمیایی است و بخار تولید شده از آنها در هنگام استفاده، برای بدن مضر است. هنگام استفاده از این مواد باید:

- از ماسک مناسب استفاده کنید.
- محل کار دارای تهویه مناسب باشد.



با مراجعه به رسانه‌های مختلف انواع روغن لحیم مونتاژ و ديمونتاژ قطعات SMD را شناسایی کنید و موارد کاربرد آنها را در یک جدول تنظیم نموده و به کارگاه ارائه کنید.

■ **پنس (Clip Forceps):** برای برداشتن قطعات SMD از ابزاری به نام پنس استفاده می‌شود. برخی از پنس‌ها با توجه به ابعاد قطعات دارای ابعاد ظریف و نوک تیزتر هستند. با توجه به محل مونتاژ یا ديمونتاژ، می‌توان از پنس سرکج یا سرصاف استفاده کرد (شکل ۸).



شکل ۸- پنس نوک تیز سرصاف و سرکج

■ **دستگاه وکیوم (Vaccum) (مکش یا ایجاد خلأ):** برای برداشتن آی‌سی‌ها یا قطعات دیگر که امکان برداشتن آنها با پنس وجود ندارد، از وکیوم یا ابزار تولید خلأ استفاده می‌شود. این وسیله، هوای بین قطعه و سر لاستیکی ابزار را می‌مکد و خلأ تولید می‌کند. این خلأ باعث چسبیدن قطعه به دستگاه وکیوم می‌شود. دستگاه وکیوم را در اصطلاح عمومی وکیوم می‌نامند. در شکل ۹ دو نمونه وکیوم را مشاهده می‌کنید.



نوعی ابزار وکیوم دستی وجود دارد که بسیار ارزان و کم‌هزینه است، با مراجعه به رسانه‌های مختلف آن را بیابید و نتیجه را به کارگاه ارائه دهید.



شکل ۹- دو نمونه دستگاه وکیوم



شکل ۱۰- سیم لاکي نمره ۰/۱

■ **سیم لاکي:** اگر هنگام دمنوتاژ، برد صدمه ببیند و قطعی در مسیر ایجاد شود، باید مدار را دوباره سیم‌کشی و تعمیر کنیم. برای این منظور می‌توانیم سیم لاکي به کار ببریم. همان‌طور که قبلاً ذکر شد طبق شکل ۱۰ جنس این سیم‌ها از مس بوده و دارای پوشش لاکي است. پوشش لاکي سیم مانع ایجاد هرگونه اتصالی با سایر قسمت‌های مدار چاپی می‌شود.



شکل ۱۱- سیم قلع نازک

■ **سیم لحیم:** سیم لحیم برای ترمیم لحیم پایه قطعات، در مونتاژ و دمنوتاژ به کار می‌رود. برای بردهای SMD از سیم لحیم‌های نازک‌تر استفاده می‌شود. زیرا حجم قلع موردنیاز هر پایه بسیار کم است. در شکل ۱۱ یک نمونه قرقره سیم لحیم نازک نشان داده شده است.

با مراجعه به رسانه‌های مختلف انواع سیم‌های نازک لاکي مورد استفاده برای ترمیم برد مدار چاپی و انواع لحیم‌های مورد استفاده برای قطعات SMD را با توجه به قطر سیم لحیم و میزان درصد قلع و سرب آنها، بیابید و نتایج را در یک جدول خلاصه کنید و نتیجه را به کارگاه ارائه دهید.

فعالیت



انتخاب و آماده‌سازی ابزار و دستگاه‌های SMD

هدف: کسب مهارت در شناخت و آماده‌سازی ابزار و دستگاه‌های SMD

مواد، ابزار و تجهیزات: هویه قلمی - هویه هوای گرم - خمیر قلع - مایع فلاکس - سیم لحیم - روغن لحیم - قلع‌کش - پایه هویه - اسفنج مخصوص پاک‌کننده نوک هویه - فیبر اوراقی SMD - لوازم التحریر - سنباده

مراحل اجرای کار:

۱ یک جدول فهرست واریسی تهیه کنید، ابزار و دستگاه‌های SMD موجود در کارگاه را در اختیار بگیرید و مشخصات فنی هریک از آنها را بنویسید.

کار عملی ۱



ردیف	نام ابزار یا دستگاه	مشخصات فنی	کاربرد
۱			
۲			
۳			
۴			
۵			
۶			

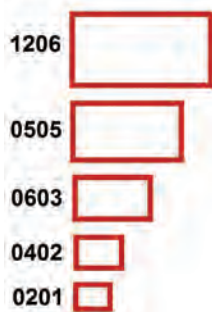
۲ با توجه به فرایند لحیم کاری (مونتاژ و دمونتاژ) قطعات SMD که توسط مربی شما در کارگاه ارائه می‌شود، فهرستی از فرایند لحیم کاری تهیه کنید.

مراحل	مونتاژ SMD	مراحل	دمونتاژ SMD
۱		۱	
۲		۲	
۳		۳	
۴		۴	
۵		۵	

خودارزیابی

- ۱ هویه‌ای که برای مونتاژ و دمونتاژ قطعات SMD استفاده می‌شود، در مقایسه با هویه‌های قلمی معمولی هم‌تراز درجه حرارت دارد.
- ۲ از سیم لحیم برای ترمیم بردهای SMD استفاده می‌شود.
- ۳ قسمت‌های مختلف هویه هوای گرم (هیتر) را شرح دهید.
- ۴ کاربرد ابزار و کیوم در لحیم کاری SMD را توضیح دهید.

استاندارد قطعات SMD



تمامی قطعات دو پایه مانند خازن، مقاومت، دیود و دیاک از نظر ابعاد (طول، عرض و قطر) از استانداردهای خاص پیروی می‌کنند. در شکل ۱۲ استانداردها نمایش داده شده است. در ادامه انواع استاندارد مقاومت‌های SMD را به‌عنوان نمونه بررسی خواهیم کرد.

شکل ۱۲- استاندارد مقاومت‌های SMD

با مراجعه به رسانه‌های مختلف، بررسی کنید که آیا اعداد نشان داده شده در شکل ۱۲ مفهوم خاصی را بیان می‌کند یا خیر؟ نتایج را به کارگاه ارائه دهید.

پژوهش



■ استانداردهای مقاومت‌های SMD

۱ مقاومت‌های نصب سطحی در ابعاد استاندارد شده مختلف ساخته می‌شوند. هر استاندارد را با کد مخصوص چهار رقمی نشان می‌دهند. در این کد، دو رقم اول طول مقاومت برحسب صدم اینچ و دو رقم بعدی پهنای قطعه را مشخص می‌کند. شکل ۱۳ و جدول ۱ استانداردهای مربوط به مقاومت‌های SMD را با مشخص کردن مقدار توان مجاز و طول و عرض نشان می‌دهد. در جدول ۱ ستون اول از سمت چپ کد استاندارد، ستون دوم مقدار حداکثر توان قابل تحمل مقاومت، ستون سوم و ستون چهارم، به ترتیب طول و عرض مقاومت را برحسب میلی‌متر نشان می‌دهد.

بارش فکری



قطعات SMD چگونه ساخته می‌شوند؟ از طریق بارش فکری موضوع را بررسی و جمع‌بندی کنید.

جدول ۱- مقاومت‌های SMD

FORM	POWER (Watt)	LENGTH (mm)	WIDTH (mm)
0402	0.063	1.0	0.5
0503	0.063	1.27	0.75
0505		1.27	1.25
0603	0.062	1.60	0.80
0705		1.91	1.27
0805	0.1	2.00	1.25
1005	0.125	2.55	1.25
1010		2.55	2.55
1206	0.25	3.2	1.6
1210	0.25	3.2	2.6
1505		3.8	1.25
2010	0.5	5.08	2.55
2208		5.72	1.90
2512	1.0	6.5	3.25
MELF		5.5	2.2
MINIMELF		3.6	1.4
MICROMELF		2.0	1.27



شکل ۱۳- مقاومت‌های SMD

با مراجعه به رسانه‌های مختلف، انواع جداول مربوط به استاندارد مقاومت‌ها، خازن‌ها، دیودها و ترانزیستورهای SMD را بیابید و نتیجه را در کارگاه به بحث بگذارید.

فعالیت



مقاومت SMD به صورت تراشه‌ای ساخته می‌شود که دو انتهای آن از داخل پایه مسطح متصل می‌شود. معمولاً لایه خارجی مقاومت SMD را قلع‌اندود می‌کنند تا تراشه برای نصب آماده باشد. طبق جدول حداکثر توان قابل تحمل مقاومت‌های استاندارد ۱۰۰۵ برابر با ۰/۱۲۵ وات است. مقاومت‌هایی با توان بالاتر با اندازه‌های بزرگ‌تر نیز وجود دارد. مثلاً مقاومت‌های استاندارد ۲۵۱۲، حداکثر توان قابل تحمل، برابر با ۱ وات دارند.

■ از قطعات دیگری که از نظر شکل و ابعاد شبیه مقاومت SMD است و در مدارهای SMD زیاد دیده می‌شود، قطعات اتصال کوتاه‌کننده مقاومت صفر اهمی است. مقاومت این قطعات صفر بوده و در عمل مقاومتی در مسیر جریان ایجاد نمی‌کند. این اتصال کوتاه‌کننده‌ها که نمونه آن در شکل ۱۴ نشان داده شده است، برای

ایجاد پل بین دو مسیر مسی در برد مدار چاپی مورد استفاده قرار می‌گیرد. روی قطعه اتصال کوتاه‌کننده‌های صفر اهمی ممکن است علامتی نباشد یا عدد ۰۰۰ بر روی آنها نوشته شده باشد.



شکل ۱۴- اتصال کوتاه‌کننده صفر اهمی / مقاومت‌های SMD

با مراجعه به رسانه‌های مختلف درباره انواع مقاومت‌های SMD استفاده شده در تجهیزات الکترونیکی تحقیق کنید و نتیجه را در کارگاه به بحث بگذارید.

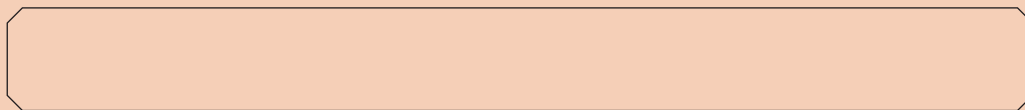
پژوهش



فعالیت



تعدادی مقاومت SMD در اختیار بگیرید و از روی ابعاد آنها، کد استاندارد هریک را مشخص کنید.



■ روش خواندن مقدار مقاومت‌های SMD

با توجه به فناوری ساخت و مقدار توان، مقاومت‌های SMD متنوع بوده و در ابعاد گوناگونی ساخته می‌شود. با توجه به ابعاد مقاومت SMD، مقدار مقاومت آن را به دو روش «مستقیم» و «کدگذاری شده» درج می‌کنند. ■ **مستقیم:** در روش مستقیم مقدار اهمی مقاومت روی آن نوشته می‌شود.

■ **کدگذاری:** به علت کوچک شدن ابعاد قطعه مقدار آن را به کد تبدیل و کد را روی قطعه حک می‌کنند. کدگذاری با سه روش به شرح زیر صورت می‌گیرد:

الف) نوارهای رنگی: مانند کد رنگی مقاومت‌های TH، هر رنگ بیانگر یک عدد است و روش آن مشابه مقاومت‌های معمولی است.

ب) کدگذاری عددی: این روش نیز تا حدودی مشابه مقاومت‌های معمولی است. در این روش مقدار مقاومت را با یک عدد سه یا چهار رقمی به شرح زیر مشخص می‌کنند.

■ **روش سه رقمی:** مانند مقاومت‌های معمولی، دو رقم اول، ارقام اصلی مقادیر مقاومت است و به تعداد رقم سوم در مقابل دو رقم اول، صفر می‌گذاریم.

مثال

$$203 = 20000 = 20 \text{ K}\Omega$$

$$100 = 10 \Omega$$

$$566 = 56 \times 10^6 = 56 \text{ M}\Omega$$



■ **روش چهاررقمی:** مانند مقاومت‌های معمولی، سه رقم اول، ارقام اصلی مقادیر مقاومت است و به تعداد رقم چهارم در مقابل سه رقم اول صفر قرار می‌دهیم.

مثال

$$2001 = 2000 = 2 \text{ K}\Omega$$

$$3260 = 326 \Omega$$

$$4753 = 475 \times 10^3 = 475 \text{ K}\Omega$$



جدول ۲- ضریب توان ده مقاومت های SMD

حرف	R	K	M
ضریب	۱	10^3	10^6

■ **عدد و حرف:** این روش به دو صورت «دو عدد و یک حرف» و «دو عدد و دو حرف» استفاده می شود. در این روش حرف اول ضریب (تعداد صفر) و حرف دوم مقدار خطا را نشان می دهد. برای نشان دادن اعشار نیز از همان حرف اول استفاده می شود. جدول های ۲ و ۳ مقادیر ضرایب و خطا را مشخص می کنند.

جدول ۳- تolerانس مقاومت های SMD

حرف	B	C	D	F	G	H	J	K	M
درصد خطا	۰/۱	۰/۲۵	۰/۵	۱	۲	۳	۵	۱۰	۲۰

مثال

$$56K = 56 \text{ K}\Omega$$

$$5M6 = 5/6 \text{ M}\Omega$$

$$K56 = 0/56 \text{ K}\Omega = 560 \Omega$$

$$R12 = 0/12 \Omega$$

$$1R0 = 1 \Omega$$

$$23KJ = 23 \text{ K}\Omega \pm 5\%$$

$$4K7M = 4/7 \text{ K}\Omega \pm 20\%$$



با مراجعه به رسانه های مختلف یک نرم افزار بیابید که بتواند رمز مقاومت های SMD را آشکار و مقدار مقاومت را رمزگذاری کند.

جستجو کنید



کار عملی ۲



انتخاب و خواندن مقادیر مقاومت های SMD

هدف: کسب مهارت در شناخت قطعه و خواندن مقادیر اهمی مقاومت های SMD
مواد، ابزار و تجهیزات: قطعات SMD - فیبر اوراقی SMD - لوازم التحریر
مراحل اجرای کار:



شکل ۱۵- نمونه برد اوراقی

۱ مقاومت های برد شکل ۱۵ یا برد اوراقی دیگر را شناسایی کنید.

۲ مقدار مقاومت های آن را تعیین کنید.

این مقاومت ها با چه استانداردی تطابق دارند؟
 از چه نوع کدهایی برای این مقاومت ها استفاده شده است؟

استانداردهای حاکم بر سایر قطعات

برای سایر قطعات SMD مانند خازن SMD، سلف، ترانزیستور و دیود نیز استانداردهایی وجود دارد که در ادامه به شرح آن می‌پردازیم.

■ **استاندارد حاکم بر خازن‌های SMD:** خازن برای ذخیره انرژی به صورت الکتروستاتیک در مدارات استفاده می‌شود. این قطعه را با حرف C نمایش داده و واحد آن فاراد (F) است. نمادهای شماتیکی خازن به صورت شکل ۱۶ می‌باشد.



ب) خازن‌های دارای قطب

الف) خازن بدون قطب

شکل ۱۶- نماد انواع خازن

خازن‌های SMD معمولاً از جنس الکتrolیتی، سرامیکی و تانتالیوم هستند. رنگ خازن‌های سرامیکی قهوه‌ای و خاکستری است. این نوع خازن‌ها قطبیت ندارند. خازن‌های تانتالیوم SMD برخلاف خازن‌های سرامیکی دارای قطب مثبت و منفی هستند. رنگ خازن تانتالیوم SMD زرد است و قطب مثبت آن را با یک خط یا علامت «+» مشخص می‌کنند.



شکل ۱۹- خازن سرامیکی SMD و معمولی

شکل ۱۸- خازن الکتrolیتی SMD

شکل ۱۷- خازن تانتالیوم SMD

■ **استاندارد سلف (سیم‌پیچ)‌های SMD:** خاصیت القایی سیم‌پیچ را با حرف L نشان می‌دهند. در نقشه‌ها، سلف با علائم شکل ۲۰ مشاهده می‌شود:



د) سلف با هسته هوا

ج) سلف با هسته آهنی

ب) سلف متغیر دارای هسته فریت

الف) سلف با هسته فریت

شکل ۲۰- نماد انواع سلف



برخی سلف‌ها دارای جهت هستند و هنگام مونتاژ و دیمونتاژ باید به چگونگی قرار گرفتن آنها روی برد دقت کرد. معمولاً جهت قرار گرفتن قطعه روی برد با یک نقطه مشخص می‌شود. سلف‌های SMD با رنگ‌های سیاه و سفید ساخته می‌شوند، (شکل ۲۱). سلف‌های معمولی که به سلف‌های مقاومتی نیز معروف هستند، مانند مقاومتهای با رنگ سیاه ساخته می‌شوند.

شکل ۲۱- سلف‌های SMD سفید و سیاه

برخی از سلف‌ها با روکش پلاستیک ساخته می‌شوند. در بعضی از بردها مجموعه‌ای از سیم پیچ‌ها در یک بسته شبیه به آی‌سی قرار گرفته‌اند. نمونه‌ای از این سلف‌ها در شکل ۲۲ مشاهده می‌شود.



شکل ۲۲- انواع سلف‌های SMD

سلف‌های مقاومتی معمولاً برای حفاظت در مقابل جریان استفاده می‌شوند. سلف‌های استوانه‌ای نیز بیشتر در مدارهای مختلف الکترونیکی به کار می‌روند. ترانس‌های SMD معمولاً مکعبی هستند، مانند ترانس‌های معمولی برای تطبیق امپدانس یا کوپلاژ و کاهش یا افزایش مقدار ولتاژ در مدار قرار می‌گیرند.

■ **استاندارد دیودهای SMD:** دیود را با حرف D نمایش می‌دهند. دیود انواع مختلفی دارد که از مهم‌ترین آنها می‌توان به دیود زener، مادون قرمز و دیود نورانی اشاره کرد. دیود معمولی برای یکسوسازی جریان و ولتاژ، دیودهای زener برای تثبیت ولتاژ و دیودهای نورانی برای تولید نور به کار می‌رود. در مدارهای الکترونیکی دیود را به صورت شکل ۲۳ نشان می‌دهند.



شکل ۲۳- نماد انواع دیودها

دیودهای SMD معمولی مانند مقاومت به رنگ سیاه بوده ولی اندازه آنها نسبت به مقاومت بزرگتر است. چند نمونه دیود SMD را در شکل ۲۴ مشاهده می‌کنید.



ج) دیود زبر SMD

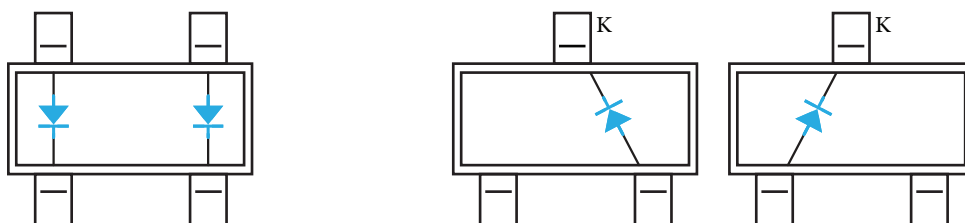
د) بسته دیودی شامل دو دیود

ب) چند نمونه دیود SMD

الف) دیود معمولی SMD

شکل ۲۴- شکل ظاهری انواع دیود SMD

برخی دیودها از نظر شکل ظاهری شبیه به ترانزیستور و آی سی هستند. یافتن پایه‌های آند و کاتد این دیودها با مولتی‌متر امکان پذیر است. اتصالات داخلی دیودهایی که بسته‌بندی آنها بیش از دو پایه دارند، در شکل‌های ۲۵ و ۲۶ مشاهده می‌شود.



شکل ۲۶- دیود در بسته دارای چهارپایه

شکل ۲۵- دیود در بسته دارای سه پایه (ترانزیستوری)

با توجه به نمونه‌هایی از دیودهای SMD که در شکل ۲۵ آورده شده است، در قالب بارش فکری به سؤالات زیر پاسخ دهید و در نهایت جمع‌بندی کنید.

- علت استفاده از بسته‌هایی با سه پایه برای یک دیود چیست؟
- روش یافتن آند و کاتد دیود در بسته‌های دیودی چگونه است؟

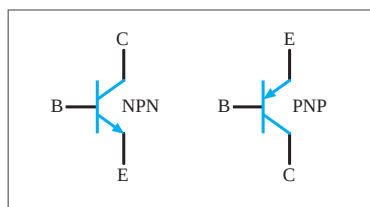
بارش فکری



جستجو کنید



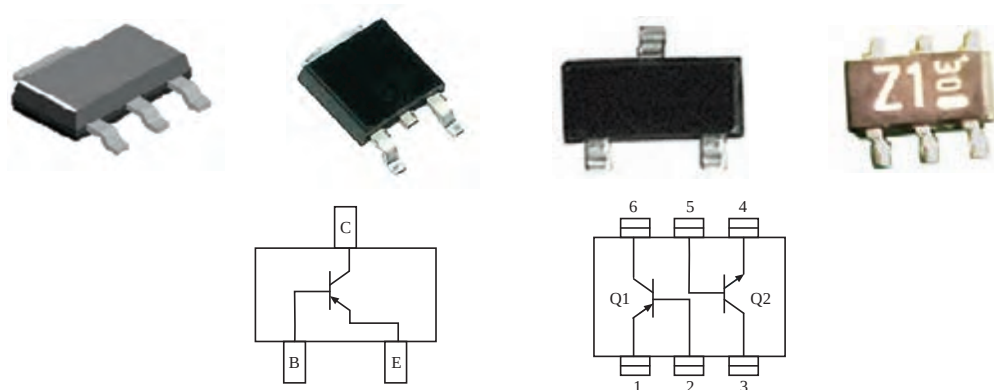
با مراجعه به فضای مجازی اطلاعات مربوط به دو نمونه از انواع دیودهای SMD را استخراج نموده و به کارگاه ارائه دهید.



■ **استاندارد ترانزیستورهای SMD:** در مدارهای الکترونیکی ترانزیستور برای کلیدزنی و تقویت کنندگی استفاده می‌شود. ترانزیستور را با حروف Q یا Tr نشان می‌دهند. نماد فنی ترانزیستور مثبت (NPN) و منفی (PNP) در شکل ۲۷ نشان داده شده است.

شکل ۲۷- نماد ترانزیستورهای مثبت و منفی

شکل ظاهری ترانزیستورهای SMD به صورت قطعاتی سه پایه یا چهارپایه به رنگ سیاه هستند. در برخی بردهای الکترونیکی، دو یا چند ترانزیستور به صورت یکپارچه در یک بسته بندی به صورت آی سی قرار می‌گیرد. شکل ظاهری چند نمونه ترانزیستور و اتصال داخلی دو نمونه ترانزیستور SMD در شکل ۲۸ نشان داده شده است.



شکل ۲۸- شکل ظاهری ترانزیستورهای SMD فشرده شده (compact) و اتصالات داخلی آن

انواع دیگر قطعات الکترونیکی مانند ترایستور و تریاک SMD نیز از نظر شکل ظاهری مشابه ترانزیستور SMD هستند. برای شناسایی این قطعات باید با توجه به شماره فنی آنها مراجعه کرد و اطلاعات مربوط به آن را از برگه اطلاعات یا دیگر منابع اطلاعاتی فنی استخراج نمود.

تذکر



باید توجه داشته باشیم که قطعات SMD به شکل رول هایی با تعداد زیاد بسته بندی می‌شوند به عنوان مثال مقاومت ها در رول های ۵۰۰۰ تایی قرار می‌گیرند برخی از قطعات SMD واقع در رول های استاندارد در شکل ۲۹ نشان داده شده‌اند.



شکل ۲۹- رول های بسته بندی انواع قطعات SMD



انتخاب و خواندن خازن‌ها، دیودها و ترانزیستورهای SMD

هدف: کسب مهارت در شناخت قطعه و خواندن مقادیر قطعات SMD
مواد، ابزار و تجهیزات: قطعات SMD - فیبر اوراقی SMD - لوازم التحریر
مراحل اجرای کار:

برد اوراقی موجود در کارگاه را در اختیار بگیرید و نوع قطعات موجود در روی برد را مشخص نموده و در جدول یادداشت کنید.

ردیف	نوع قطعه	ویژگی‌ها، استاندارد و مقدار
۱		
۲		
۳		
۴		
۵		

شست‌وشوی برد

قبل از هر مرحله مونتاژ و یا دیمونتاژ قطعات باید برد مدار چاپی تمیز و از وجود هرگونه گردوغبار و رطوبت پاک شود. برای این کار به مواد و ابزارهایی مانند اسپری خشک، فرچه نرم (قلم‌مو یا برس مویی نرم) و پمپ باد نیاز دارید. برای تمیز کردن برد باید ابتدا برد را توسط فرچه نرم یا پمپ باد از گردوغبار پاک کنید. سپس برد را به وسیله اسپری خشک شست‌وشو دهید. در نهایت با فرچه نرم تمام قسمت‌ها را به‌طور کامل سرویس کنید. در صورت عدم دسترسی به اسپری، یا هزینه بالای آن، می‌توانید از حلال‌های دیگر مانند تینر فوری (تینر ۱۰۰۰۰) نیز استفاده نمود. به‌جای فرچه نرم می‌توانید از مسواک نیز استفاده کنید. شکل ۳۰ نحوه آغشته کردن فرچه به تینر و تمیز کردن برد را نشان می‌دهد.



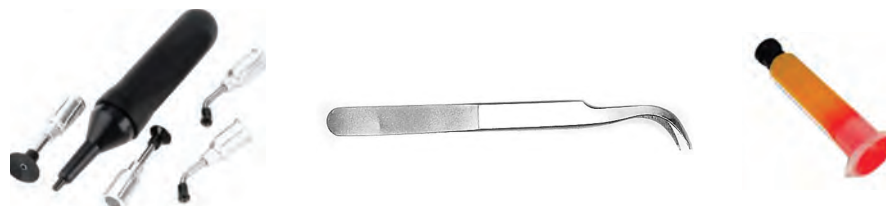
شکل ۳۰- شست‌وشوی برد

مواد تمیزکننده مانند اسپری و تینر بسیار خطرناک و آتش‌زا هستند، هنگام کار با این مواد کلیه نکات ایمنی را رعایت کنید.



خودارزیابی

۱ تجهیزاتی که برای برداشتن قطعه دمونتاز شده از روی برد استفاده می‌شود، را در شکل ۳۱ علامت بزنید:



شکل ۳۱- تجهیزات

۲ رفع عیب قطعی فیبر مدارهایی که هنگام دمونتاز به وجود آمده است، با استفاده انجام می‌شود.

۳ با مراجعه به جداول مرتبط مقدار واقعی مقاومت‌های زیر را بیابید:

الف) $4K\Omega$ (ب) $1R0$ (ج) $76B$ (د) $09R$

۴ مقاومت‌های سبز از مشکی دقیق‌تر هستند.

صحيح □ غلط □

۵ خازن‌های تانتالیوم SMD اکثراً به رنگ هستند.

الف) آبی (ب) سبز (ج) خاکستری (د) زرد

۶ مقاومت‌های خطای ۵ درصد دارند.

الف) مشکی (ب) سبز (ج) آبی (د) گزینه ب و ج

۷ مقاومت $7K4J$ معادل مقاومتی با اهم است (با مراجعه به جدول).

الف) $74K\Omega \pm 5\%$ (ب) $74K\Omega \pm 10\%$ (ج) $7/4K\Omega \pm 5\%$ (د) $7/4K\Omega \pm 10\%$

۸ مقاومت با کد ۲۳۰۲ چند اهم است؟

الف) ۲۳۰ کیلو اهم (ب) ۲۳ کیلو اهم (ج) $230/2$ کیلو اهم (د) ۲۳۰ اهم

۹ $K96$ معادل کدام مقاومت است؟

الف) ۹۶ کیلو اهم (ب) ۹۶ اهم (ج) ۹۶۰ اهم (د) $9/6$ کیلو اهم

۱۰ از وکیوم برای استفاده می‌شود.

الف) حمل قطعه دمونتاز شده (ب) تسهیل لحیم کاری (ج) سرویس برد (د) برداشتن قلع اضافه

۱۱ ICهایی که پایه‌های آنها در کف و زیر قطعه است را چه می‌نامند؟

الف) BGA (ب) QFTP (ج) BGY (د) گزینه الف و ج

پیاده‌سازی (دمونتاژ) قطعات SMD (مقاومت، سلف و خازن)

برای پیاده‌سازی انواع قطعات SMD، ابتدا باید میزان دما و میزان شدت وزش هوای خروجی دستگاه هوای گرم را با توجه به استانداردهای تعریف شده برای قطعه تنظیم کنیم. میزان دما با توجه به جنس برد و نوع قطعه تعیین می‌شود. برای کارهای معمولی دمای بین ۳۰۰ تا ۴۰۰ درجه مناسب است. میزان خروجی هوای گرم نیز باید در حدی تنظیم شود که قطعات از روی برد به اطراف پرتاب نشوند. پس از تنظیم دستگاه، قبل از شروع کار دمونتاژ، محدوده کار روی برد را با استفاده از فرچه تمیز کنید و با مایع مناسب و استاندارد شست‌وشو دهید تا هرگونه گرد و غبار اضافی از روی برد پاک شود. سپس مقداری خمیر فلاکس روی پایه‌های قطعه اضافه کنید تا دمونتاژ آسان‌تر شود.

مطابق شکل ۳۲، هیتر را به آرامی و به صورت عمود روی پایه‌های قطعه بگیرید و حرارت دهید. برای جدا کردن آی‌سی‌ها از روی برد، دقت کنید تا به تمام پایه‌های آی‌سی حرارت یکنواخت داده شود. بعد از مدتی کوتاه، با پنس یک ضربه بسیار ضعیف به قطعه بزنید. اگر تمام پایه‌های قطعه از برد جدا شده باشد می‌توانید آن را به وسیله پنس یا دستگاه وکیوم بردارید.

از آنجا که اکثر کارهای عملی این پودمان را روی بردهای اوراقی انجام خواهیم داد برای حفظ قطعات بردهای



شکل ۳۲- دمونتاژ برد با تکنولوژی SMT

اوراقی باید قطعات دمونتاژ شده را روی برد دوباره مونتاژ کنید برای این کار ابتدا سطح مورد نظر را به فلاکس آغشته کنید. سپس لایه‌ای از قلع را به کمک هوای پدها قرار دهید به این کار قلع اندود کردن پدها گوئیم. سپس قطعه را در محل مربوطه قرار داده و به وسیله هوای روی تک‌تک پایه‌های قطعه حرارت دهید به این ترتیب عمل لحیم‌کاری انجام می‌شود. توضیحات تکمیلی در واحد یادگیری دوم ارائه خواهد شد.

■ جدا کردن مقاومت، سلف و خازن‌های SMD از روی برد فیبر مدار چاپی

چون این قطعات تنها دو پایه دارند، جداسازی آنها به کمک هیتر از روی برد فیبر مدار چاپی به آسانی صورت می‌گیرد.

برای برداشتن این قطعات باید پایه‌ها را به طور متناوب حرارت دهید. یعنی طبق شکل ۳۳، پس از حرارت دادن یک پایه به مدت کوتاه، به پایه دیگر حرارت دهید. حرارت دادن به پایه‌ها را تا ذوب شدن لحیم هر دو پایه ادامه دهید. هنگام دمونتاژ این قطعات باید دقت کنید که میزان هوای خروجی از هیتر زیاد نباشد. در غیر این صورت قطعه به اطراف پرتاب می‌شود. به خازن‌های الکترولیت نباید حرارت زیاد داد، زیرا سبب انفجار آنها می‌شود.



شکل ۳۳- دمونتاز خازن سرامیکی و الکترولیت SMD

کار عملی ۴



پیاده‌سازی مقاومت، خازن و سلف SMD

هدف: کسب شایستگی در دمونتاز مقاومت، خازن و سلف SMD از روی برد مدار چاپی
مواد، ابزار و تجهیزات: هویه هوای گرم - خمیر فلاکس - پنس - برد اوراقی
مراحل اجرای کار:

- ۱ قسمتی از برد مدار چاپی اوراقی که می‌خواهید روی آن کار کنید را شست‌وشو دهید.
- ۲ چند نمونه مقاومت با استانداردهای مختلف را از روی برد جدا کنید. آیا در پیاده‌سازی آنها تفاوتی مشاهده می‌شود؟ تنظیمات هیتر و زمان لازم برای دمونتاز همه آنها یکسان است؟ نتیجه را شرح دهید.
- ۳ چند نمونه خازن الکترولیت، سرامیکی و تانتالیوم را از روی برد جدا کنید. دمونتاز هریک از این خازن‌ها چه تفاوتی با یکدیگر دارند؟ شرح دهید.

حرارت زیاد باعث انفجار خازن‌های الکترولیتی و ذوب شدن قسمت‌های پلاستیکی آنها می‌شود. هنگام دمونتاز باید به این نکته توجه داشته باشید و آن را در کلیه فرایندها رعایت کنید.

نکته ایمنی

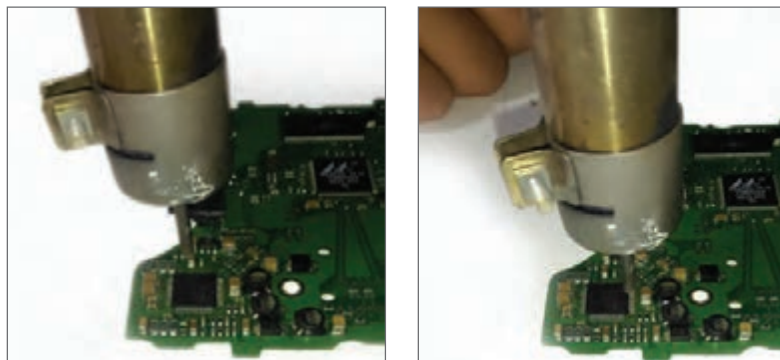


دمونتاز انواع آی‌سی‌ها

■ آی‌سی‌های خرچنگی (مربعی پایه‌دار - چهارطرفه)

تعداد پایه‌های این نوع آی‌سی‌ها بسیار زیاد است و از چهار طرف آی‌سی خارج می‌شود، به همین دلیل به آن، آی‌سی خرچنگی می‌گویند. هنگام کار با این آی‌سی‌ها باید خیلی دقیق عمل شود. با توجه به استانداردهای تعریف شده، درجه حرارت هیتر را روی دمای ۴۰۰ تا ۵۰۰ (یا درجه ۵) و درجه هوای خروجی را روی درجه ۳ یا ۴ تنظیم کنید. پس از تنظیم هیتر و تمیز کردن محدوده کار، پایه‌های آی‌سی را به خمیر فلاکس آغشته کنید. سپس پایه‌های آی‌سی را به‌طور کاملاً یکنواخت حرارت دهید، پس از مدتی با مشاهده حالت جدا شدن

آی‌سی، می‌توانید آن را با پنس یا وکیوم جدا کنید. دقت کنید پنس با پایه‌های آی‌سی برخورد نکند. آی‌سی را از دونقطه از دو انتهای قطری آن با پنس بگیرید و آن را بردارید. در صورتی که نتوانید آی‌سی را با پنس بردارید، باید از دستگاه وکیوم استفاده کنید. شکل ۳۴ مراحل از اجرای کار را نشان می‌دهد.



شکل ۳۴- دمونتاژ آی‌سی‌های چهارطرفه (خرچنگی)

■ آی‌سی‌های BGA

از آنجا که پایه‌های این نوع از آی‌سی‌ها درکف و زیر قطعه قرار دارد، مشاهده نمی‌شوند. باید قبل از برداشتن، محل قرار گرفتن آن را روی برد دقیق یادداشت کنید، شکل ۳۵ تصویر یک نمونه آی‌سی پایه از کف یا BGA را نشان می‌دهد. نشانه‌گذاری با توجه به قطعات اطراف آی‌سی کمک خوبی برای اجرای صحیح این کار است. لازم است جهت قرارگرفتن آی‌سی (ترتیب پایه‌ها) نیز مشخص شود. معمولاً جهت آی‌سی با نوشته‌ها و علائم روی بُرد و روی آی‌سی مشخص می‌شود. این مورد را نیز باید یادداشت کنید.



شکل ۳۵- آی‌سی‌های BGA

■ برداشتن کانکتورهای SMD با پوشش پلاستیکی

برخی از قطعات اتصال‌دهنده‌ها (کانکتورها) دارای پوشش پلاستیکی هستند (شکل ۳۶). برای جدا کردن این کانکتورها می‌توانید طبق شکل ۳۷ با توجه به نوع پایه‌های کانکتور به‌طور مستقیم به پایه‌ها حرارت دهید. در صورت عدم امکان حرارت مستقیم می‌توانید طبق شکل ۳۸ از زیر برد حرارت دهید و قطعه را بردارید. روش دیگر، قرار دادن پارچه خیس روی قطعه و حرارت دادن آن است. روش اصلی و ایمن برای برداشتن این نوع قطعات استفاده از هیتر مادون قرمز می‌باشد که تنها قلع را ذوب می‌کند و به قسمت‌های پلاستیکی آسیبی نمی‌رساند.



شکل ۳۸- حرارت از زیر برد



شکل ۳۷- حرارت دادن مستقیم



شکل ۳۶- کانکتور پلاستیکی SMD



پیاده‌سازی یک آی‌سی BGA

هدف: کسب شایستگی در دمونتاز یک آی‌سی از برد SMD

مواد، ابزار و تجهیزات: هویه هوای گرم - سشوار صنعتی - پنس - برد اوراقی
مراحل اجرای کار:

یک آی‌سی BGA را یک بار توسط هیتر و یک بار توسط سشوار صنعتی دمونتاز کنید. هر دو حالت را با هم مقایسه نموده و در قالب یک گزارش در کلاس ارائه دهید.

هیترهای صنعتی

برای جدا کردن قطعات حساس SMD مانند آی‌سی‌های کارت گرافیک در لپ‌تاپ‌ها یا قطعاتی که روی بردهای چند لایه قرار گرفته‌اند نیاز به دقت بسیار بالا برای دمونتاز دارند. برای دمونتاز این قطعات از هیترهای صنعتی استفاده می‌شود. در شکل ۳۹، یک نمونه از این نوع هیتر را مشاهده می‌کنید. این هیتر از قسمت‌های زیر تشکیل شده است:

■ **دوربین مدار بسته:** این دوربین برای نشان دادن قطعه موردنظر به کار می‌رود. با استفاده از دوربین مدار بسته می‌توانید مکان دقیق پایه‌های قطعات را روی برد مدار چاپی ببینید. در شکل ۴۰ صفحه نمایش (مانیتور) یک نمونه از این نوع دوربین را مشاهده می‌کنید.

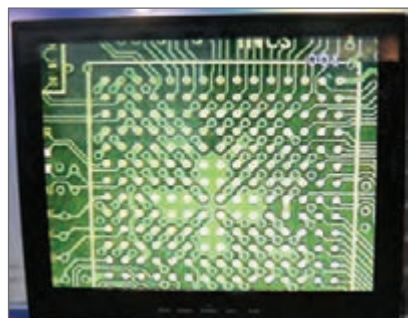


شکل ۴۰- مانیتور و تصویر دوربین مدار بسته

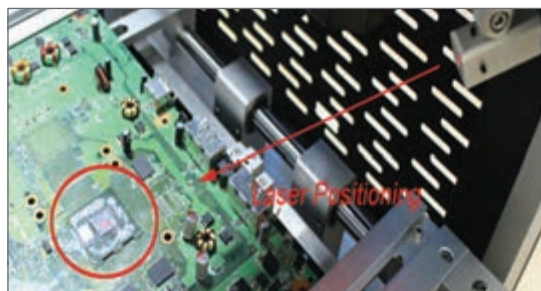


شکل ۳۹- هیتر صنعتی

همچنین هنگام دمونتاز توسط این دستگاه نیز می‌توان با دقت خیلی زیاد چگونگی قرار گرفتن قطعه‌ها و اتصال‌های آنها را مورد بررسی قرار داد (شکل ۴۱).

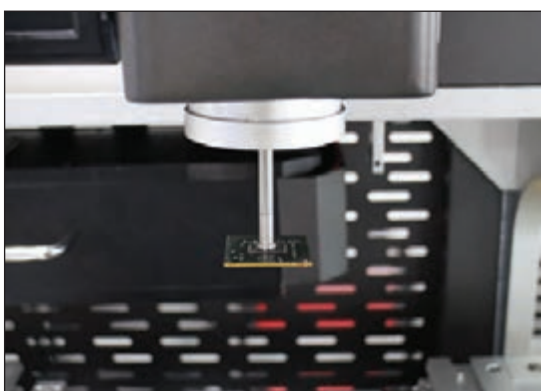


شکل ۴۱- مشاهده وضعیت پایه‌ها



شکل ۴۲- لیزر

■ **لیزر:** برای نشان دادن و تمرکز دستگاه روی قطعه موردنظر برای دمونتاز به کار می‌رود (شکل ۴۲).



شکل ۴۳- وکیوم هیتر صنعتی

■ **هیتر:** قسمت حرارتی برای دمونتاز قطعات و ذوب لحیم‌ها به کار می‌رود.

■ **وکیوم:** برای برداشتن قطعه دمونتاز شده از روی برد مورد استفاده قرار می‌گیرد. در شکل ۴۳ عمل وکیوم کردن دستگاه برای برداشتن یک قطعه SMD نشان داده شده است.

تهیه فهرست قیمت: ابتدا فهرستی از قطعات، مواد، ابزار و دستگاه‌های موردنیاز تهیه کنید سپس با مراجعه به رسانه‌ها و مراکز مختلف قیمت آنها را استخراج کنید و در جدول ۴ بنویسید. نتیجه را به صورت گروهی به بحث بگذارید.

فعالیت



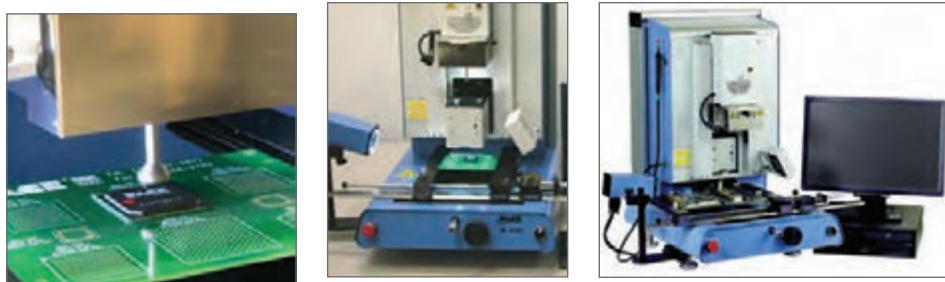
جدول ۴- برآورد قیمت قطعات، مواد، ابزار و دستگاه‌های موردنیاز برای دمونتاز قطعات SMD

ردیف	مشخصات فنی	قیمت

نام و نام خانوادگی تهیه کننده..... تاریخ

هیتروهای مادون قرمز (IR)

این هیتروها با تابش اشعه مادون قرمز، قلع پایه‌ها را ذوب می‌کنند. هیتروهای مادون قرمز برای جداسازی قطعات خاص مانند آی‌سی‌های BGA و یا BGY به کار می‌رود. در شکل ۴۴ یک هیترو مادون قرمز مشاهده می‌شود.



شکل ۴۴- هیترو مادون قرمز (IR)

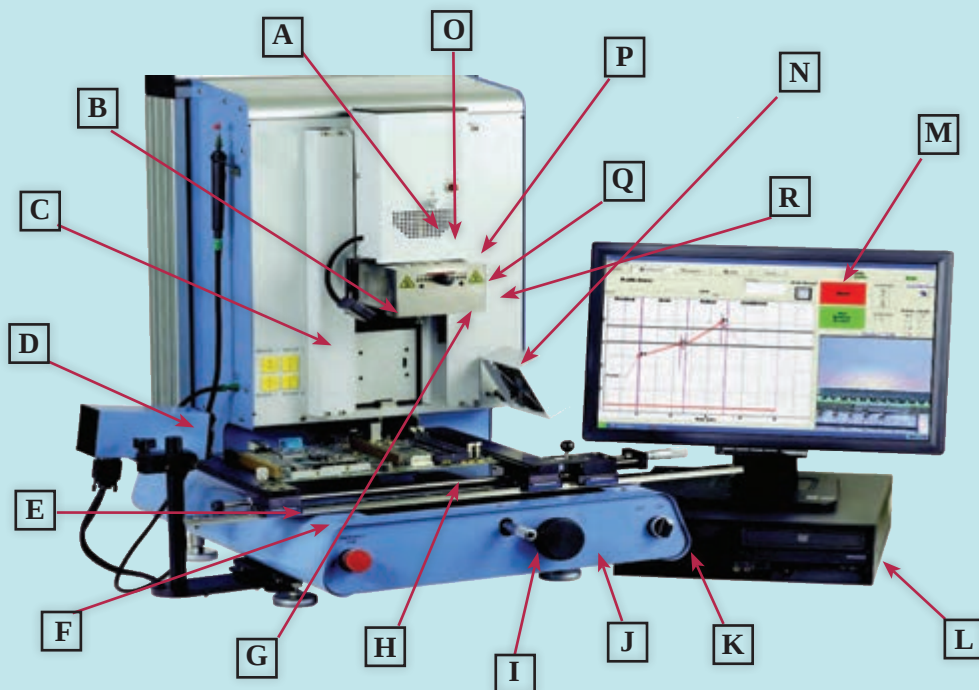
تعیین اجزای مختلف یک هیترو صنعتی

هدف: کسب شایستگی برای شناختن اجزای یک دستگاه هوای گرم (Heater) مادون قرمز (IR)

مواد، ابزار و تجهیزات: لوازم التحریر، دفترچه راهنمای دستگاه هوای گرم

مراحل اجرای کار:

۱ شکل ۴۵ قسمت‌های مختلف یک هیترو مادون قرمز را نشان می‌دهد. با توجه به دفترچه راهنمای دستگاه که در ادامه آمده است و فیلم نمایش داده شده، نام قسمت‌های مختلف را در کنار شکل ۴۵ مشخص کنید.



شکل ۴۵- هیترو مادون قرمز IR

کار عملی ۶



یک نمونه راهنمای کاربرد هیتر مادن قرمز به زبان انگلیسی

- | | |
|-------------------------|---|
| A. Reflow Head | Contains the top-side heater and moves up and down via an electric motor that is controlled through the software. The reflow head is cutched to control downward force. |
| B. IR sensor | Used to measure temprature on top of component as the primary means of developing a profile. |
| C. Sensor Inputs | The sensor inputs are K-type thermo-couples. Measured temperatures are displayed through the PC software in real time for use in making profile graphs. |
| D. Sodr-Cam | Used for "real-time" viewing of solder ball melting. |
| E. Board Holder | Fine adjustment of both the X and Y direction is achieved by using the adjusting knobs on the end of the holder for x and on the front of the machine for Y. The right side of the holder is spraing loaded to hold the PCB securely. |
| F. Botton Heater | Powerful IR heating source used to provide bottom-side board heating and is adjustable to a height of 1.5". |
| G. Emergency Off Switch | In case an emergency shut down is necessary, press this button. |
| H. Optics Housing | Contains the camera and beam splitter (prism). The housing extents and retracts automatically during operation and the lights for the optics will turn on/off automatically when the housing is extended/retrasted. |
| J. Micrometer Dials | Precision "x" and "y" axis board alignment micrometer dials for accurate positioning and aligning of component to PCB. |
| K. Bottom Heater | Used to adjust the height of the heater with respect to adjustment Knob the |
| Height Adjust | bottom-side board surface. Can be raised to a maximum of 1.5" height. |
| L. On/Off Switch | Used to turn the system on or off. When turning off the system, always turn off the PC using the windows interface first. When starting the system, always turn on the IR 3000 before starting the PC software. |

M. Keyboard	Used to enter information into software.
N. Mouse	Used to enter information into software.
P. Display Monitor	Display PC Software.
Q. Cooling Fan	The component and PCB are cooled by the coling fan, and can be activated automatically after the reflow cycle is compelete or operated manually.
R. Aperture Adjustment Knobs	Used to change the size and/or shape of IR pattern to the size of the component being reworked.
S. Theta Dial	Enables 360 degree rotation of component for alignment to PCB.
T. IR Heater Indicator	Neon heater lamp indicates topside IR heater activity.

بازدید

جلسه بازدید از یک مرکز که بردهای الکترونیکی با قطعات SMD را تهیه می کنند، برنامه ریزی شود و از قبل اهداف بازدید برای هنرجویان مشخص شود و از آنان خواسته شود که گزارش نسبتاً جامعی از فرایند تولید مدار چاپی با استفاده از هیتر صنعتی و هیتر مادون قرمز ارائه کنند.

کار عملی ۷



الگوی پرسش عملی

دمونتاژ مقاومت

- برد اوراقی را سرویس کنید.
- مقاومت های برد را شناسایی کنید.
- هیتر را تنظیم کنید.
- پایه های حداقل سه مقاومت را به خمیر فلاکس آغشته کرده و آن را از برد جدا کنید.
- گزارش عملکرد و مشکلات خود را در سه سطر بنویسید.

ارزشیابی شایستگی دمونتاز SMD

نام و نام خانوادگی		شماره ملی		تاریخ ارزشیابی:		نوبت:
کد حرفه	۸۲۱۲۱۲	حرفه:	مونتاژکاران تجهیزات و وسایل برقی و الکترونیکی	کد پیمانه/پودمان	۸۲۱۲۱۲۰۳۱	استاندارد عملکرد: با رعایت اصول ایمنی، با انتخاب ابزار و مواد و تجهیزات مناسب قطعات SMD را از برد اوراقی پیاده سازی کند.
کد وظیفه شغل	۸۲۱۲۱۲۰۳	وظیفه شغل:	ساخت مدارهای دیجیتال و میکروکنترلر	سطح صلاحیت	۲	
کد کار	۸۲۱۲۱۲۰۳۱۱	کار	دمونتاز SMD	سطح شایستگی	مهارت	

ردیف	مرا حل کار	شرایط انجام کار (ابزار، مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص ها/داوری /نمره دهی)	نتایج ممکن	نمره کسب شده
	بررسی سلامت برد، پدها، مسیرها و آماده سازی محیط کار	زمان کافی، محیط کار استاندارد، ابزار SMD، برد اوراقی سالم	مشاهده مستقیم، چک لیست بررسی فنی	بررسی کامل، شناسایی قطعات و انتخاب ابزار	۳	
				عدم دقت کافی در شناسایی قطعه و انتخاب ابزار	۲	
				عدم شناسایی صحیح قطعه یا انتخاب اشتباه ابزار	۱	
۲	شناسایی قطعه و تنظیم دما و باد هیتر	کارگاه الکترونیک، زمان تعیین شده، برد سالم، و ابزار SMD موجود و هیتر کالیبره با تنظیمات دمایی استاندارد	مشاهده فرایند لحیم کشی/ جداسازی، بررسی پدها پس از انجام کار	تنظیم دقیق و جداسازی	۳	
				انحراف دمایی جزئی و آسیب سطحی ترمیم پذیر	۲	
				آسیب شدید به برد و قطعات	۱	
۳	خارج کردن قطعه SMD بدون آسیب به پدها	بررسی ظاهری پایه های قطعه و تست سلامت با مولتی متر کالیبره در محیط روشن	نتایج تست مولتی متر و مشاهده سلامت پایه ها	بررسی کامل فنی و تشخیص صحیح سلامت قطعه	۳	
				تست با خطا و بررسی ناقص	۲	
				عدم انجام تست یا تشخیص اشتباه سلامت قطعه	۱	
۴	کنترل سلامت قطعه، دسته بندی اصولی و تکمیل گزارش کار	کارگاه روشن و مرتب، برد سالم، مولتی متر کالیبره، دیتاشیت، ظروف و برجسب، زمان مشخص و نوشت افزار گزارش	چک لیست مشاهده ای تنظیم هیتر و جابجایی قطعه، نتایج مولتی متر، نظم ظروف و برجسب ها و صحت گزارش کار	تشخیص صحیح، تنظیم هیتر، تست، بایگانی و گزارش کامل	۳	
				تشخیص صحیح، تنظیم با اصلاح جزئی، تست، بایگانی و گزارش ناقص	۲	
				آسیب به قطعه، خطا در تشخیص/تست، بایگانی و گزارش ناقص	۱	
	شایستگی های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:	کارگاه الکترونیک مجهز به ابزار و دستگاه های SMD، تجهیزات حفاظت فردی و قطعات مورد نیاز، انجام فعالیت های کارگاهی مطابق با دستورالعمل های ایمنی و بهداشت حرفه ای	مشاهده رفتار هنرجو در حین انجام فعالیت، میزان رعایت اصول ایمنی و نظم، فعالیت های گروهی، مدیریت زمان و مسئولیت پذیری	رعایت کامل ایمنی، بهداشت و محیط زیست، رفتار حرفه ای، نظم و مسئولیت پذیری بالا	۳	
				رعایت نسبی ایمنی و بهداشت با خطاهای جزئی، رفتار کاری مناسب	۱	
ارزشیابی کار (شایستگی انجام کار)						☐ بلی
						☐ خیر

توضیحات:

۱- معیار شایستگی انجام کار:

- کسب حداقل نمره ۲ در بازرسی دقیق برد، تنظیم صحیح هیتر، جاسازی بدون آسیب و تست و بایگانی منظم با مستند سازی و تکمیل گزارش کار (مراحل بحرانی)

- کسب حداقل نمره ۲ از بخش شایستگی های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش

- کسب حداقل میانگین ۲ از مراحل کار

۲- تعریف سطوح شایستگی: صرف نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه ای انجام می شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار

اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه ریزی و تحلیل، پاسخ گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب یابی، هدایت و راهنمایی دیگران.

واحد یادگیری ۲: مونتاژ SMD

آیا تا به حال فکر کرده‌اید

- چگونه می‌توانیم قطعات SMD را مونتاژ کنیم؟
- اتصال یک قطعه SMD بعد از مونتاژ چه ویژگی‌هایی باید داشته باشد؟
- برای مونتاژ قطعات SMD از چه موادی استفاده می‌شود؟
- برای مونتاژ قطعات SMD چه روش‌هایی وجود دارد؟
- برای مونتاژ قطعات SMD از چه ابزارها و تجهیزاتی استفاده می‌شود؟
- مونتاژ قطعات الکترونیکی SMD با مونتاژ قطعات مجزا (discrete) چه تفاوت‌هایی دارد؟
- چرا تقریباً در همه دستگاه‌های الکترونیکی به روز از قطعات SMD استفاده شده است؟

پس از اتمام این واحد یادگیری، فراگیرنده باید علاوه بر کسب مهارت در انتخاب روش‌های مونتاژ این نوع قطعات و به کارگیری ابزارهای موردنیاز، مهارت لازم را در اجرای مونتاژکاری قطعات نصب سطحی کسب کند. یادآور می‌شود که مونتاژ قطعات SMD با سه روش دستی، اتوماتیک و نیمه‌اتوماتیک صورت می‌گیرد. در این واحد یادگیری تأکید بر مونتاژ قطعات به صورت دستی داریم. نکات ایمنی و بهداشت و مهارت‌های غیرفنی مانند مسئولیت‌پذیری، کار گروهی، نظم در کار، توجه به محیط‌زیست و اخلاق حرفه‌ای با توجه به اهمیتی که دارد باید در تمام موارد رعایت شود.

استاندارد عملکرد

نصب قطعات SMD روی برد اوراقی براساس استانداردهای تعریف شده

مواد، ابزارها و تجهیزات

هویه قلمی، هویه هوای گرم، هویه پنسی، خمیر قلع (لحیم)، مایع فلاکس، سیم لحیم، روغن لحیم، قلع کش، پایه هویه، اسفنج مخصوص پاک‌کننده نوک هویه، فیبر اوراقی SMD، لوازم التحریر، سنباده، پنس و دستمال نخی



دو مورد از مزایای استفاده از قطعات SMD را نام ببرید.



شکل ۴۶- کارگاه مونتاژ دستی قطعات الکترونیکی

در این کارگاه‌ها تمام کارها توسط اپراتور و به صورت دستی انجام می‌شود. شخصی که در این نوع کارگاه‌ها کار می‌کند را مونتاژ کار قطعات الکترونیکی می‌نامند. در فرایند مونتاژ، قطعات را به صورت دستی روی برد می‌گذارند و عمل لحیم کاری را با هویه یا هویه هوای گرم (داغ) به صورت دستی انجام می‌دهند. شکل ۴۶ یک نمونه مونتاژ دستی قطعات الکترونیکی را نشان می‌دهد.



شکل ۴۷- مونتاژ دستی بردهای الکترونیکی

در شکل ۴۷ نمونه دیگری از مونتاژ دستی برد الکترونیکی تلفیقی TH و SMD را ملاحظه می‌کنید.



شکل ۴۸- مونتاژ دستی قطعات SMD

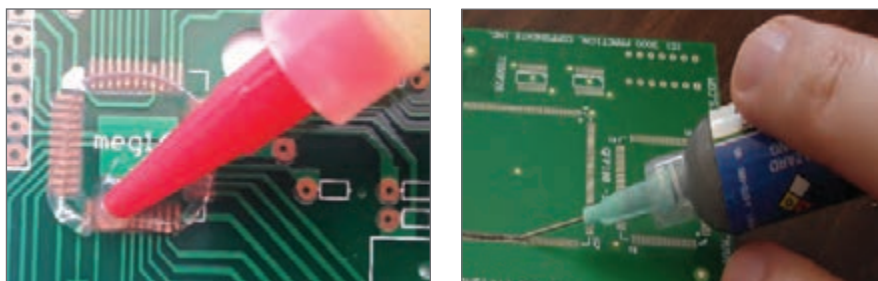
لحیم کاری دستی قطعات SMD را با دو روش استفاده از هویه معمولی و هویه هوای گرم و خمیر قلع انجام می‌دهند، شکل ۴۸ تصویر یک نمونه مونتاژ دستی قطعات SMD را نشان می‌دهد.

■ لحیم کاری قطعات SMD با استفاده از هویه

در این روش ابتدا سطوح موردنظر را به فلاکس یا روغن لحیم آغشته می‌کنند. سپس لایه‌ای از قلع یا خمیر قلع روی آن می‌گذارند. سپس قطعه را در محل مربوطه قرار می‌دهند در انتها به وسیله هویه مخصوص SMD و لحیم یا خمیر لحیم پایه‌ها را قلع‌اندود می‌کنند. محل قرار گرفتن پایه‌های قطعات باید یکنواخت و بدون اتصال به یکدیگر باشد و حالت برجسته و براق پیدا کند. در روش دیگر اجرا، محل قرار گرفتن قطعات روی برد مدار چاپی را هم قلع‌اندود می‌کنند، سپس قطعه را روی برد می‌گذارند و عمل لحیم کاری را انجام می‌دهند.

مراحل اجرای کار به شرح زیر است:

■ آغشته کردن محل لحیم کاری به ماده فلاکس یا روغن لحیم (شکل ۴۹).



شکل ۴۹- اضافه کردن مایع فلاکس در محل قرارگیری قطعه SMD

■ قلع اندود کردن محل پایه ها با استفاده از هویه، لحیم مفتولی یا خمیر قلع (لحیم) (شکل ۵۰).

■ به وسیله حلال مناسب پاک کننده، مانند بنزین، الکل، تینر یا اسپری های مخصوص پاک کردن برد، محل قطعه و اطراف آن را با مسواک شست و شو دهید. سپس قبل از اینکه حلال تبخیر شود از ابرهای اسفنجی استفاده کنید و آن را با فشار زیاد روی سطح برد بکشید تا سطح لحیم کاری شده از روغن لحیم و ذرات ریز پاک شود (شکل ۵۱).
■ با کمک پنس، قطعه یا IC مورد نظر را در محل خود تنظیم کنید.
■ در فرایند لحیم کاری ابتدا یک یا دو پایه قطعه را با استفاده از هویه در محل خود کمی فشار دهید تا در قلع ذوب شده و آوسی در محل خود محکم شود.



شکل ۵۰- قلع اندود کردن محل پایه ها با هویه و سیم لحیم

■ بعد از باز بینی و کسب اطمینان از اینکه IC در محل صحیح خود قرار دارد، سایر پایه ها را به همین ترتیب لحیم کنید.

■ صحت تنظیم دقیق در محل نصب قطعه بسیار مهم است و معمولاً اگر در مرحله ای که یک یا دو پایه را محکم می کنید قطعه به درستی تنظیم نشده باشد می توانید با استفاده از قلع با قطر کم دوباره لحیم کاری کنید.

■ اگر احتمالاً دو پایه به صورت ناخواسته به هم بچسبند، به کمک قلع کش مناسب می توانید لحیم های اضافی را بردارید. همچنین با استفاده از فتیله لحیم آغشته به روغن لحیم و هویه، می توانید لحیم های اضافی را از بین دو پایه بردارید.

فیلم مربوط به مونتاژ کاری قطعات SMD به وسیله هویه را مشاهده کنید.



شکل ۵۱- پاک کردن سطح زیر قطعه به کمک ابرهای اسفنجی

مونتاژ تعدادی مقاومت SMD بر روی برد

هدف: کسب مهارت در مونتاژ قطعات SMD با استفاده از هویه قلمی.

مواد، ابزار و تجهیزات: فیبر اوراقی، مفتول لحیم، سنباده نرم، پایه هویه، هویه قلمی و اسفنج مخصوص پاک کردن نوک هویه، مایع فلاکس، مقاومت های SMD استاندارد ۱۲۰۶، ابر اسفنجی، مسواک، تینر یا حلال مخصوص پاک کردن برد، قلع کش، سیم قلع کش، فتیله لحیم، دستمال نخی

کار عملی ۱



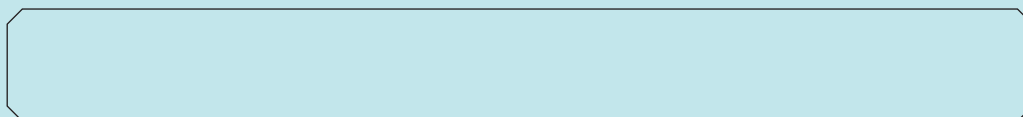
در این مرحله لازم است از بُرد اوراقی فعالیت عملی پودمان قبل، یعنی دمونتاز قطعات SMD استفاده کنید.



از آنجا که در واحد یادگیری قبل با تجهیزات مربوط به لحیم کاری آشنا شده‌اید، با نکات ایمنی و بهداشتی مربوط به آن نیز آشنا هستید. لذا ضرورت دارد کلیه موارد ایمنی و بهداشتی را به‌طور کامل رعایت کنید.

مراحل اجرای کار:

- ۱ ابزار مورد نیاز را انتخاب و آماده کنید.
- ۲ برای لحیم کاری با استفاده از هویه قلمی و از سیم لحیم با قطر کم استفاده کنید.
- ۳ برچسب روی قرقره سیم لحیم را بررسی کنید و مشخصات آن را بنویسید.



- ۴ دستگاه هویه قلمی را بررسی کنید و از سالم بودن آن مطمئن شوید.
- ۵ تعدادی مقاومت SMD با استاندارد ۱۲۰۶ را در اختیار بگیرید.
- ۶ ابعاد مقاومت‌های SMD را اندازه‌گیری کنید و مشخصات فنی آنها را در جدول ۵ بنویسید.

جدول ۵

ردیف	طول	عرض	استاندارد
R۱			
R۲			
R۳			
R۴			

۷ یک قطعه برد اوراقی مدار الکترونیکی که در کار عملی واحد یادگیری اول تعدادی از مقاومت‌های آن را دمونتاز کرده‌اید در اختیار بگیرید و آن را مورد بازبینی قرار دهید و استاندارد مقاومت‌هایی که دمونتاز شده‌اند را شناسایی کنید.

۸ تعدادی از مقاومت‌ها را در محل‌های مناسب مونتاژ کنید. برای مونتاژ به ترتیب زیر عمل کنید:

- مقداری مایع فلاکس یا روغن لحیم در محل‌های مورد نظر قرار دهید.
- با استفاده از هویه قلمی مخصوص قطعات SMD مقداری لحیم را در محل‌های مورد نظر قرار دهید و پایه‌ها را با هویه قلع‌اندود کنید.
- با استفاده از تینر یا حلال مخصوص پاک کردن برد، محل قطعه و اطراف آن را با مسواک شست‌وشو دهید، سپس به‌وسیله ابر اسفنجی مخصوص قبل از اینکه حلال تبخیر شود با فشار زیاد روی سطح برد بکشید تا سطح از روغن لحیم و ذرات ریز پاک شود.

■ به کمک پنس مقاومت یا قطعه موردنظر را در محل خود تنظیم کنید و با هویه در محل خود لحیم کنید. اگر احتمالاً دو پایه به صورت ناخواسته به هم بچسبند به کمک قلع کش مناسب می توانید لحیم های اضافی را بردارید. همچنین با استفاده از فتیله لحیم آغشته به روغن لحیم و هویه می توانید لحیم های اضافی را بردارید.

۹ ■ با قلع کش و فتیله لحیم برای برداشتن لحیم اضافی تمرین کنید تا تسلط کامل را به دست آورید.

۱۰ ■ یک قطعه مقاومت SMD را طبق استانداردهای تعریف شده روی برد لحیم کنید.

۱۱ ■ تمام مراحل فوق را برای مقاومت های دیگر نیز تکرار کنید.

۱۲ ■ گزارش کوتاهی از مونتاژ مقاومت های SMD به کمک هویه را بنویسید.

■ لحیم کاری با استفاده از خمیر لحیم و هویه هوای گرم

در این روش به جای قلع مفتولی از خمیر قلع و به جای هویه از هویه هوای گرم استفاده می شود. خمیر لحیم ترکیبی از قلع و مایع روغن لحیم (فلاکس) است که با گرم کردن آن مایع روغن لحیم (فلاکس) تبخیر شده و قلع باقی می ماند. در واقع روغن لحیم سطح کار را تمیز و عملیات لحیم کاری را آسان می کند. شکل ۵۲ نمونه ای از روغن مخصوص و شکل ۵۳ نمونه ای از خمیر لحیم را نشان می دهد.



شکل ۵۳- نمونه هایی از خمیر لحیم



شکل ۵۲- روغن فلاکس

■ مراحل اجرای لحیم کاری با هویه هوای گرم

■ برای لحیم کاری با روش هویه هوای گرم باید ابتدا سطح برد را به وسیله مواد پاک کننده تمیز کنید، سپس لایه ای نازک از خمیر لحیم روی تمامی پدها قرار دهید. ضخامت این لایه باید در حدی باشد که بتوانید سطح PCB و پدها را ببینید (شکل ۵۴). این مرحله را اندود کردن برد با خمیر لحیم می نامند.



شکل ۵۴- قرار گرفتن خمیر قلع روی تمامی پدها



خمیر لحیم یک ماده سمی است، مراقب باشید که این خمیر با پوست شما تماس پیدا نکند. در صورت تماس احتمالی و ناخواسته بلافاصله محل آلوده شده را با آب و صابون بشویید. بعد از اتمام کار حتماً دست و صورت خود را کاملاً شست و شو دهید.

■ بعد از مرحله تمیزکاری و اضافه کردن خمیر لحیم قطعه را به کمک پنس در مکان خود به گونه‌ای قرار دهید که هر پایه بر روی پد مخصوص خود قرار گیرد. قطعه را باید نگه‌دارید تا در اثر جریان هوای گرم جابه‌جا نشود، توجه داشته باشید که پنس در لحیم‌کاری و هوادهی بسیار داغ خواهد شد. بنابراین باید از پنسی استفاده کنید که دستگیره غیرفلزی داشته باشد.



شکل ۵۵- قرار دادن قطعه در محل خود

■ نازل هویه را در فاصله ۸ سانتی‌متری از قطعه به مدت ۲۰ تا ۴۰ ثانیه نگه‌دارید تا هوادهی انجام شود. در اثر هوای گرم خمیر کاملاً ذوب می‌شود. پس از اطمینان از اینکه قطعه دقیقاً در محل خود قرار دارد، نازل را به تدریج دور کنید. قطعه را آن‌قدر نگه‌دارید تا محل اتصال‌های لحیم‌کاری شده، خنک و محکم شود (شکل ۵۵).

■ پس از اتمام لحیم‌کاری محل‌های اتصال را بازرسی کنید. در این بازبینی لازم است تمام پدها با دقت بررسی شود. پایه‌ها باید محکم لحیم شده باشد و نباید قلع اضافی سبب اتصال دو پایه به هم شود. در صورت وجود قلع اضافی به کمک قلع‌کش و فتیله لحیم قلع اضافی را بردارید.

چه روش‌ها و ابزارهایی برای مونتاژ قطعات SMD وجود دارد. در مورد آنها جست‌وجو کنید و نتیجه را از طریق بارش فکری در کارگاه به بحث بگذارید.



مونتاژ تعدادی مقاومت SMD بر روی برد اوراقی با استفاده از خمیر لحیم و هویه هوای گرم

هدف: کسب مهارت مونتاژ قطعات SMD با استفاده از هویه هوای گرم و خمیر لحیم.

مواد، ابزار و تجهیزات: فیبر اوراقی، خمیر لحیم، هویه هوای گرم، مایع فلاکس، مقاومت‌های SMD استاندارد ۱۲۰۶، ابر اسفنجی، مسواک، تینر یا حلال مخصوص پاک کردن برد، قلع‌کش، سیم قلع‌کش، پنس با دسته پلاستیکی و دستمال نخی

مراحل اجرای کار:

۱ ابزار موردنیاز را انتخاب و آماده کنید.

۲ برچسب روی خمیر لحیم را بررسی کنید و مشخصات آن را بنویسید.

۳ راهنمای کاربرد هویه هوای گرم را مطالعه کرده و مشخصات فنی آن را به‌طور خلاصه بنویسید.

۴ تعدادی مقاومت SMD با استاندارد ۱۲۰۶ را در اختیار بگیرید.

۵ ابعاد و مشخصات مقاومت‌های SMD را بررسی کنید و استاندارد هر کدام را بنویسید.

- ۶ برد اوراقی مدار الکترونیکی که قبلاً با آن کار کرده بودید را مورد بازبینی قرار دهید و استاندارد مقاومت‌هایی که دمونتاز شده را مشخص نمایید.
- ۷ تعدادی از مقاومت‌ها را در محل‌های مناسب مونتاژ کنید. برای مونتاژ به ترتیب زیر عمل کنید:
 - ابتدا سطح برد را به وسیله مواد پاک‌کننده تمیز کنید.
 - لایه‌ای نازک از خمیر لحیم روی تمامی پدهای محل‌هایی که قرار است لحیم‌کاری شود قرار دهید.
 - ضخامت این لایه باید به اندازه‌ای باشد که بتوانید سطح مسی PCB و پدها را ببینید.
 - قطعه را به کمک پنس در محل خود قرار دهید و محکم نگه دارید.
 - نازل هویه را در فاصله ۸ سانتی‌متری از قطعه به مدت ۲۰ تا ۴۰ ثانیه بگیرید تا خمیر لحیم کاملاً ذوب شود پس از اطمینان از اینکه قطعه دقیقاً در محل خود قرار دارد نازل را به تدریج دور کنید.
 - اگر احتمالاً دو پایه به صورت ناخواسته در اثر خمیر اضافی به هم چسبیده‌اند با قلع‌کش مناسب یا فتیله لحیم، لحیم اضافی را بردارید تا اتصال برطرف شود.
- ۸ تمام مراحل فوق را برای مقاومت‌های دیگر نیز تکرار کنید.
- ۹ گزارش کوتاهی از مونتاژ مقاومت‌های SMD به کمک هویه هوای گرم و خمیر لحیم را بنویسید.

دستگاه‌های مونتاژ تمام خودکار (اتوماتیک)

در این روش تمام فرایند مونتاژ به صورت خودکار توسط ماشین انجام می‌شود. در این فرایند فایل مدار طراحی شده با نرم‌افزارها روی رایانه دستگاه تمام اتوماتیک بارگذاری می‌شود (شکل ۵۶). برد مدار چاپی نیز با توجه به راهنمای استفاده از دستگاه در محل مخصوص قرار می‌گیرد. قطعات که قسمت عمده آن نوع SMD است از طریق درگاه‌های تغذیه قطعه روی دستگاه قرار می‌گیرند. قطعات معمولاً روی لایه‌های مقوا یا کاغذ به شکل نواری نصب شده و به صورت حلقه (رول) بسته‌بندی شده‌اند. این رول‌ها در محل‌های مخصوص روی دستگاه قرار می‌گیرد. برد مدار چاپی که روی یک نقاله قرار گرفته است وارد قسمت پرینتر خمیر لحیم می‌شود. دستگاه ابتدا خمیر لحیم را در محل‌های مسی مدار، چاپ می‌کند سپس برد روی نقاله حرکت کرده و وارد قسمت قطعه‌گذاری (pick and place) می‌شود. در این قسمت قطعات SMD به وسیله ربات از روی نوارهای قطعات برداشته می‌شود و در محل صحیح خود قرار می‌گیرد و پس از نصب کامل قطعات، نقاله برد را به سمت کوره دستگاه می‌برد. در کوره خمیر لحیم ذوب شده و عمل لحیم‌کاری صورت می‌گیرد و پس از اتمام کار برد به آرامی به بیرون هدایت می‌شود.



آیا دستگاه‌های کوچک تمام خودکار برای اجرای لحیم کاری وجود دارد؟ با مراجعه به رسانه‌های مختلف اطلاعات مورد نیاز را استخراج و به کارگاه ارائه دهید.



شکل ۵۶- دستگاه تمام خودکار مونتاژ قطعات SMD

کارگاه‌های مونتاژ نیمه اتوماتیک

در این کارگاه‌ها قسمت اعظم فرایند عملیات به صورت دستی اما با کمک برخی ماشین آلات صورت می‌گیرد. معمولاً بردهای تولیدی در این کارگاه‌ها دارای ترکیبی از قطعات SMD و TH هستند. مراحل مونتاژ یک مدار الکترونیکی در کارگاه‌های نیمه اتوماتیک به شرح زیر است:

- اندود کردن برد با خمیر لحیم
- قطعه گذاری قطعات SMD
- ذوب خمیر لحیم و لحیم کاری قطعات SMD
- مونتاژ دستی قطعات TH
- کنترل کیفی یا QC

■ مرحله اول: اندود کردن برد با خمیر لحیم

برای مونتاژ قطعات SMD باید با خمیر لحیم، برد را خمیر اندود کنید. خمیر لحیم به رنگ خاکستری است. این خمیر باید قبل از قطعه گذاری روی تمام پدهایی که قطعه سوار می‌شود قرار گیرد. خمیر اندود کردن برد مدار چاپی به دو صورت انجام می‌شود.



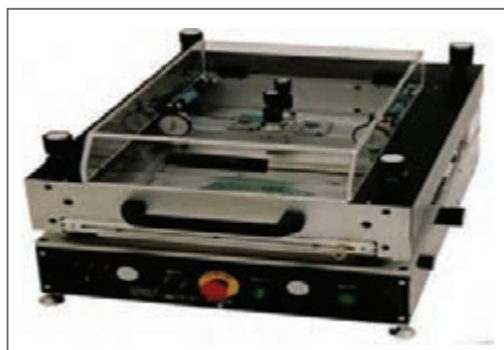
شکل ۵۷- خمیر اندود کردن لحیم به کمک شابلون

■ با استفاده از شابلون: روش شابلون شبیه چاپ سیلک است. در این روش از یک شابلون فلزی استفاده می‌شود. روی شابلون، محل پایه قطعاتی که باید مونتاژ شود برش خورده است. پس از ثابت کردن شابلون روی PCB شکل ۵۷ خمیر لحیم را روی آن قرار می‌دهند و تمام منافذ و حفره‌هایی که مشخص کننده محل پایه‌هاست را با خمیر لحیم پر می‌کنند. به این عمل خمیر اندود کردن می‌گویند.



در مورد انواع روش‌های ساخت شابلون تحقیق کنید و نتیجه را در قالب یک گزارش به کلاس ارائه دهید.

با استفاده از دستگاه پرینتر خمیر لحیم: در روش پرینتر خمیر لحیم از یک دستگاه خودکار برای خمیراندود کردن برد استفاده می‌شود. در این روش شابلون به صورت اتوماتیک روی مدار چاپی تنظیم می‌شود و خمیر لحیم به صورت اتوماتیک توسط دستگاه روی مدار چاپی قرار می‌گیرد (شکل ۵۸).

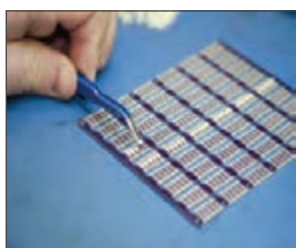


شکل ۵۸- دستگاه پرینتر خمیر لحیم

■ مرحله دوم: قطعه‌گذاری قطعات SMD

بعد از اینکه خمیر در محل‌های مناسب قرار گرفت، قطعه‌گذاری می‌کنیم. قطعه‌گذاری کاری بسیار ساده ولی بسیار حساس است. به دو روش می‌توانیم قطعه‌گذاری قطعات SMD را انجام دهیم. در روش اول قطعات SMD را با دست و با استفاده از یک پنس با دقت در محل‌های خود چیدمان می‌کنیم.

برای انجام این کار دو نوع پنس وجود دارد. یکی پنس معمولی است که قطعه توسط آن برداشته می‌شود و بدون اعمال حرارت در محل خود قرار می‌گیرد. پس از استقرار همه قطعات، برد را در داخل کوره قرار می‌دهند. دومین نوع هویه پنسی است که علاوه بر نگه‌داشتن قطعه، بعد از جاگذاری می‌تواند با اعمال حرارت عمل لحیم‌کاری را نیز انجام دهد. در شکل ۵۹ پنس معمولی و هویه پنسی نشان داده شده است.



ب) چند نمونه پنس



الف) هویه پنسی

شکل ۵۹- استفاده از پنس برای قطعه‌گذاری

در شکل ۵۹ یک نمونه پنس را مشاهده می‌کنید که با اتصال به قطعه SMD و دستگاه مولتی‌متر، می‌تواند کمیت‌های مربوط به قطعه SMD را اندازه بگیرد. با مراجعه به منابع معتبر بررسی کنید آیا مولتی‌متر پنسی وجود دارد؟ نتیجه را در قالب یک گزارش به کلاس ارائه دهید.





توجه داشته باشید که در بردهای الکترونیکی بزرگ، به خاطر تکراری بودن و خسته شدن از فرایند مونتاژ، بهتر است برای تولید انبوه بردها از روش جای گذاری به کمک دستگاه خودکار Pick And Place استفاده کرد.

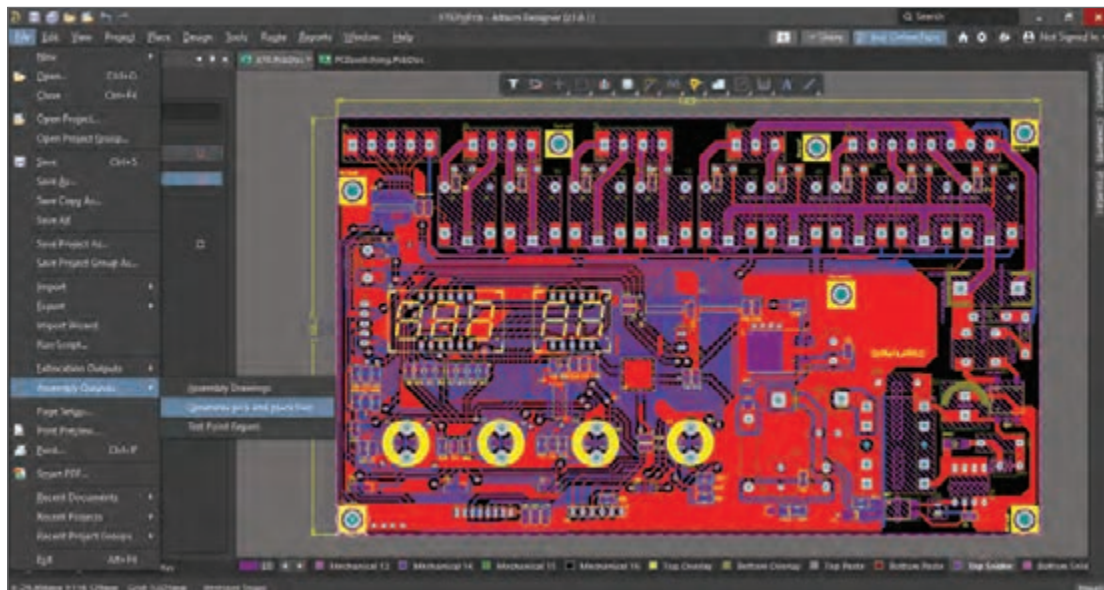
شکل ۶۰- پنس قابل اتصال به مولتی متر

■ **دستگاه خودکار Pick And Place:** این دستگاه در واقع یک ربات است که با استفاده از مکش هوا قطعات را برمی دارد و در جای خود قرار می دهد. برای راه اندازی اولیه دستگاه باید راهنمای کاربرد دستگاه را با دقت مطالعه کرد و فرایند تنظیمات آن را در طی یک دوره کارآموزی کوتاه مدت فرا گرفت تا برنامه ریزی دستگاه و اجرای تنظیمات اولیه، جای گذاری و محکم کردن برد PCB و قرار دادن رول های قطعات برد در محل خود با دقت و به ترتیب صحیح انجام شود. بعد از انجام تنظیم ها و راه اندازی دستگاه ها شما می توانید با سرعت بسیار بالایی شروع به چیدمان قطعات برد با تعداد تولید آن با تیراژ بسیار نمایید. استفاده از دستگاه Pick And Place برای تولید انبوه با صرفه است و برای تولید برد با تعداد کمتر، مثلاً روزی ۲۰ عدد با روش قطعه گذاری دستی مقرون به صرفه تر و سریع تر است. در شکل ۶۱ نمونه هایی از دستگاه پیک آند پلیس را مشاهده می کنید.



شکل ۶۱- نمونه هایی از دستگاه Pick And Place

جهت تنظیم دستگاه Pick And Place مونتاژ کار باید موقعیت قطعات SMD را به دستگاه مشخص کند. برای این کار باید فایل PCB طراحی شده در نرم افزار آلتیوم دیزاینر را باز کرده و گزارش موقعیت قطعات را در فایل اکسل دریافت کند. برای این کار مسیر file/Assembly Outputs/Generate pick and place files را مطابق شکل ۶۲ کلیک نمایید.



شکل ۶۲- نحوه دریافت فایل موقعیت قطعات

فایل اکسل ایجاد شده در آلتیوم دیزاینر شامل اطلاعات موقعیت X و Y قطعه، اسم قطعه، نوع قطعه، footprint و زاویه قرارگیری قطعه روی برد می باشد. شکل ۶۳ نمونه فایل اکسل ایجاد شده توسط آلتیوم دیزاینر را نشان می دهد.

Designator	Comment	Layer	Footprint	Center-X(mm)	Center-Y(mm)	Rotation	Description
R1	220R	TopLayer	RES_1206_(3216X08L)	67.1322	14.3028	270	resistor 1/4 wat
Q5	C945	TopLayer	SOT23_N	100.1522	33.1978	90	NPN Silicon Epitaxial Planar Transistor
Q4	C945	TopLayer	SOT23_N	31.9786	10.2108	270	NPN Silicon Epitaxial Planar Transistor
Q3	C945	TopLayer	SOT23_N	118.1354	36.83	90	NPN Silicon Epitaxial Planar Transistor
Q2	C945	TopLayer	SOT23_N	27.8638	29.9212	0	NPN Silicon Epitaxial Planar Transistor
Q1	C945	TopLayer	SOT23_N	89.408	68.326	0	NPN Silicon Epitaxial Planar Transistor
R14	10K	TopLayer	RES04817X04N	45.6946	63.1952	90	RES 1206
U1	L7805	TopLayer	D2PAK	46.1177	8.7884	90	LINEAR REGULATOR 5.0V
U5	ULN2003A	TopLayer	SO-16	74.4828	53.4212	0	Seven Darlington Array
U6	ULN2003A	TopLayer	SO-16	77.8102	18.1406	0	Seven Darlington Array
U4	ATmega32-1	TopLayer	My_44A_M	45.6946	50.9524	270	AVR 32K Bytes of In-System Programmable Flash Memory
C4	100nF	TopLayer	SMD_CAP_1206	89.4334	64.9224	0	Bipolar Capacitor
C3	100nF	TopLayer	SMD_CAP_1206	43.9166	15.9512	180	Bipolar Capacitor
R26	1K	TopLayer	RES_1206_(3216X08L)	97.1042	29.6672	90	RES 1206
R22	1K	TopLayer	RES_1206_(3216X08L)	113.2332	37.719	0	RES 1206
R27	1K	TopLayer	RES_1206_(3216X08L)	35.0266	12.9736	270	RES 1206
R21	1K	TopLayer	RES_1206_(3216X08L)	22.5806	33.2486	180	RES 1206
R12	10K	TopLayer	RES_1206_(3216X08L)	52.8262	33.7312	0	RES 1206
R11	10K	TopLayer	RES_1206_(3216X08L)	52.7812	36.1442	0	RES 1206
R4	2R7	TopLayer	RES_1206_(3216X08L)	108.3056	68.453	0	RES 1206
R5	10K	TopLayer	RES_1206_(3216X08L)	52.7812	67.0052	90	RES 1206

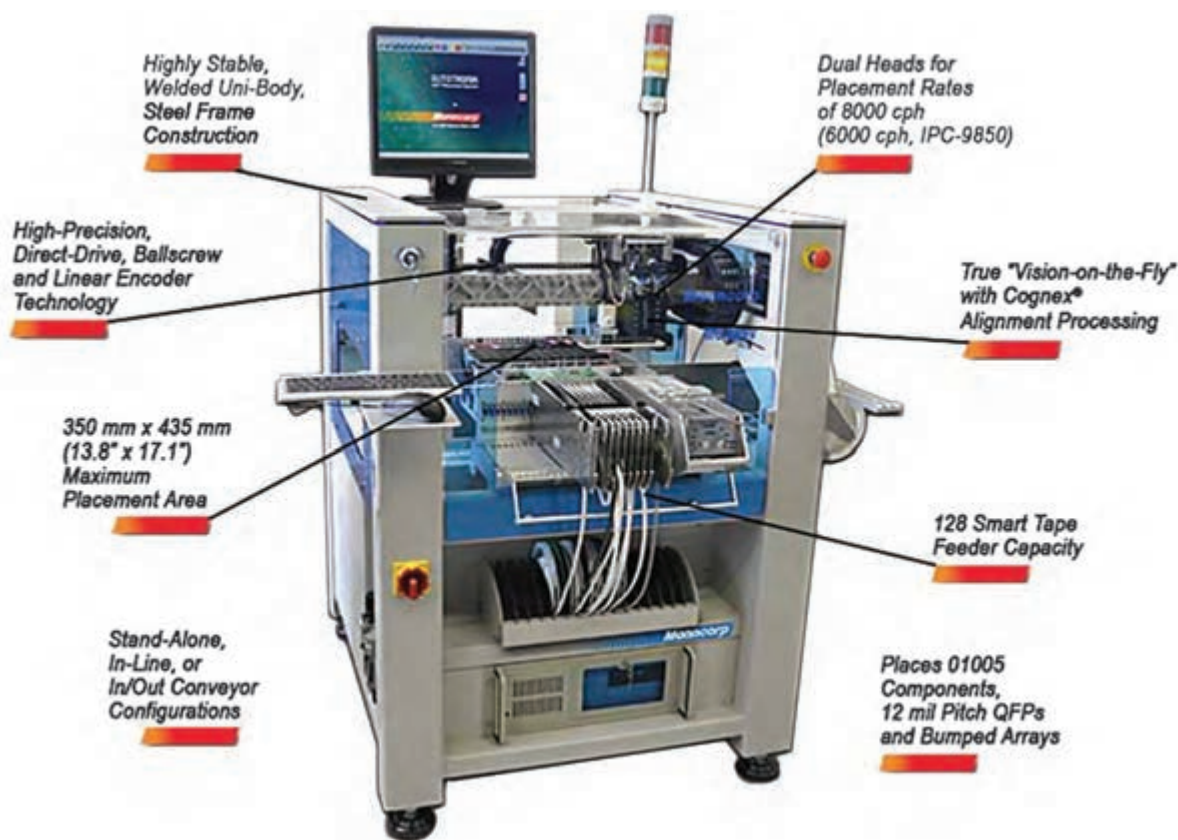
شکل ۶۳- فایل اکسل ایجاد شده توسط آلتیوم دیزاینر

[illegible][illegible]

38



با هم‌فکری اعضای گروه خود، مطالب نوشته شده به زبان اصلی در شکل ۶۵ را که مربوط به عملکرد دستگاه مونتاژ است، به فارسی ترجمه کنید و نتیجه را در کلاس ارائه دهید.



شکل ۶۵- نمونه‌هایی از دستگاه Pick And Place و اجزای تشکیل دهنده آن



در مورد انواع دستگاه‌های Pick And Place تحقیق نموده و نتیجه را به کارگاه ارائه دهید.

■ مرحله سوم: ذوب خمیر لحیم

بعد از قطعه‌گذاری به صورت دستی یا به وسیله دستگاه باید خمیر لحیم ذوب شود تا عمل لحیم کاری اتفاق بیفتد. ذوب کردن خمیر لحیم در کوره ذوب لحیم صورت می‌گیرد. این دستگاه با استفاده از امواج مادون قرمز دمای بردها را تا ۲۵۰ درجه سانتی‌گراد بالا می‌برد. پس از ذوب شدن خمیر لحیم دما را به صورت کنترل

شده پایین می‌آورد. واحد کنترل دما در زمان پایین آمدن درجه حرارت، از انبساط ناگهانی و جابه‌جایی قطعات و ایجاد ترک در محل‌های لحیم‌کاری شده جلوگیری می‌کند. شکل ۶۶ یک نمونه دستگاه کوره ذوب خمیر لحیم را نشان می‌دهد.



شکل ۶۶- دستگاه کوره جهت ذوب کردن خمیر لحیم

کوره‌های مادون قرمز در انواع مختلف و با امکانات و توان‌های متفاوت ساخته می‌شود. انتخاب توان کوره با توجه به نیاز و حجم تولید صورت می‌گیرد. پس از سرد شدن کوره برد آماده می‌شود و می‌توانید آن را از کوره خارج کرده و مورد بهره‌برداری قرار دهید. بعضی بردها دوطرفه هستند یعنی در هر دو طرف برد قطعه SMD قرار می‌گیرد. در این موارد ابتدا طرفی را که تعداد قطعات کمتری دارد مونتاژ می‌کنند. سپس طرف دیگر برد را خمیراندود کنید و مراحل را تکرار نمایید. توصیه می‌شود قبل از اینکه برد را دوباره در کوره قرار دهید طرف اول برد که لحیم‌کاری شده است را با استفاده از چسب مخصوص بپوشانید. چسب زدن مانع جابه‌جا شدن قطعات در زمان گرم شدن برد می‌شود.



شکل ۶۷- لحیم‌کاری دستی قطعات DIP با هویه و سیم لحیم

■ مرحله چهارم: لحیم‌کاری قطعات TH

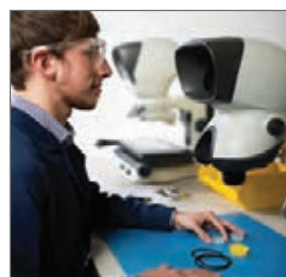
پس از لحیم شدن قطعات SMD قطعات TH باید به صورت دستی چیدمان شده و به وسیله هویه لحیم‌کاری شوند (شکل ۶۷). در تولیدات انبوه این نوع بردها از وان قلع برای لحیم‌کاری قطعات TH نیز می‌توان استفاده کرد.



شکل ۶۸- با استفاده از مانیتور برای قطعات SMD

■ مرحله پنجم: آزمایش و کنترل کیفی (QC) برد

در تولید با حجم بالا برای بازرسی بردها از دستگاه‌های خودکار استفاده می‌شود. این دستگاه‌ها کیفیت لحیم کاری، اتصالات نابجا و مونتاژ اشتباه قطعات را به سرعت تشخیص می‌دهند و به کاربر اعلام می‌کنند. یادآوری می‌شود که در تولید با حجم کم، برای QC بردها نیازی به این دستگاه‌ها وجود ندارد، با بازرسی چشمی و کمک گرفتن از لوپ‌های دستی و آزمایش تک‌تک بردها عمل QC صورت می‌گیرد (شکل ۶۸). نمونه‌ای از لوپ دیجیتال برای بازبینی چشمی در شکل ۶۹ نشان داده شده است.



شکل ۶۹- از طریق بازرسی چشمی برای قطعات SMD

در مورد نحوه مونتاژ بردهای تجهیزات الکترونیکی محیط اطراف خود تحقیق نموده و نتیجه را به کارگاه ارائه دهید.

تحقیق کنید



بازدید: از یک کارگاه مونتاژ خودکار بردهای الکترونیکی بازدید کنید و گزارش جامعی از چگونگی تولید برد ارائه دهید.

فعالیت



ساخت پروژه ۱- ساخت یک چشمک‌زن پلیسی با قطعات SMD:

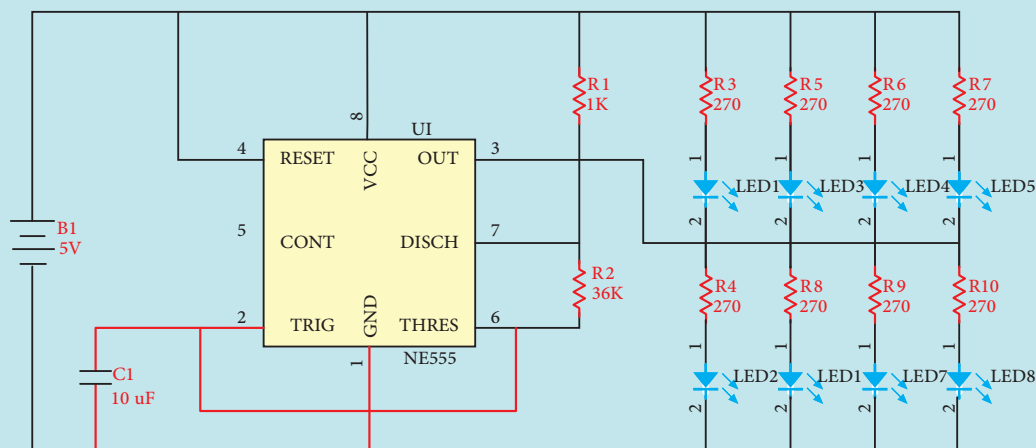
پروژه الکترونیک تعریف شده برای هنرجو شامل طراحی و نصب قطعات SMD بر روی برد مدار چاپی است. هنرجو باید توانایی راه‌اندازی و تست عملکرد برد را با دقت انجام دهد. این پروژه به منظور ارزشیابی شایستگی‌های فنی، لحیم کاری و عیب‌یابی قطعات SMD هنرجو طراحی شده است.

۱ مدار شماتیک و PCB شکل ۷۰ را در نرم‌افزار آلتیوم دیزاینر طراحی کنید.

۲ در این مدار از یک IC تایمر NE555 استفاده شده است. مقدار فرکانس IC از طریق خازن و مقاومت تنظیم می‌شود. برای تنظیم فرکانس معمولاً مقدار خازن C_1 را ثابت در نظر می‌گیرند، مقدار مقاومت‌های R_1 و R_2 را تغییر می‌دهند.

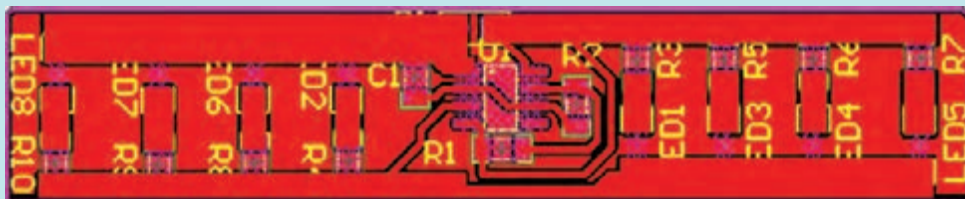
کار عملی ۳





شکل ۷۰- مدار شماتیک چشمک زن پلیسی به کمک آی سی NE555

۳ PCB مدار را طبق آموخته‌های سال دهم چاپ نمایید. در چاپ PCB حداکثر دقت لازم را انجام دهید، زیرا برای طراحی از قطعات SMD استفاده شده است. نمونه مدار PCB طراحی شده در شکل ۷۱ نشان داده شده است.



شکل ۷۱- مدار PCB چشمک زن پلیسی به کمک آی سی NE555

۴ مدار را مونتاژ نموده و تست کنید.

۵ برای مونتاژ دو نوع LED با رنگ‌های آبی و قرمز استفاده کنید.

۶ بهتر است با توجه به مشخصه LEDهای استفاده شده مقدار مقاومت آنها را محاسبه کنید تا نورهای یکسانی در هر دو رنگ داشته باشید. شکل ۷۲ کارکرد مدار نمونه را نشان می‌دهد.

۷ گزارش کاملی از فرایند اجرای پروژه ارائه دهید.



شکل ۷۲- مدار نمونه چشمک زن پلیسی به کمک آی سی NE555



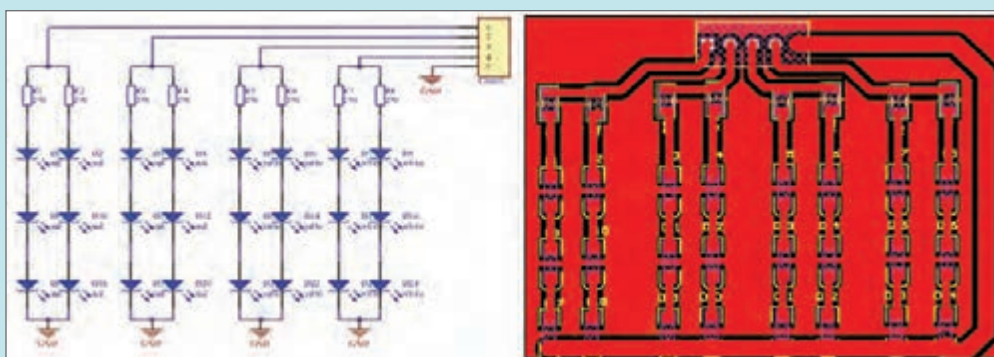
ساخت پروژه ۲- طراحی و ساخت چراغ کامل عقب خودرو:

هدف از اجرای این پروژه طراحی و ساخت چراغ عقب خودرو شامل راهنما با LEDهای زرد، چراغ خطر و چراغ عقب با LEDهای قرمز و چراغ دنده عقب با LEDهای سفید.



۱ شماتیک و PCB مدار شکل ۷۳ را در نرم افزار آلتیوم دیزاینر طراحی کنید

۲ با استفاده از اطلاعات سال دهم PCB را روی مدار چاپی چاپ کنید.
توجه: هنرجویان عزیز می توانند آرایش چراغهای عقب هر کدام از خودروها را به سلیقه خود روی برد چیدمان نمایند.



شکل ۷۳

خودارزیابی

- ۱ کدام یک از روشهای مونتاژ قطعات SMD نیاز به مفتول لحیم دارد؟
- ۲ سرعت مونتاژ در حجم انبوه قطعات SMD بیشتر است یا قطعات مجزا؟ چرا؟
- ۳ کارگاههای مونتاژ قطعات الکترونیکی به لحاظ تجهیزات به چند دسته تقسیم می شوند؟
- ۴ انواع روشهای مونتاژ دستی قطعات SMD را نام ببرید.
- ۵ روش مونتاژ به کمک هویه هوای گرم و خمیر قلع را توضیح دهید.
- ۶ دستگاه Pick And Place چه کاری انجام می دهد؟ با مراجعه به راهنمای کاربرد، مشخصات نمونه ای از آن را بنویسید.
- ۷ چرا استفاده از دستگاه Pick And Place در بسیاری از مواقع مقرون به صرفه نیست؟
- ۸ هویه پنسی چه کاربردی دارد؟ با مراجعه به کاتالوگ مشخصات یک نوع هویه پنسی را بنویسید.

تعمیق و تثبیت یادگیری

- ۱ در کارگاههای مونتاژ دستی از کدام یک از ابزارهای زیر استفاده می شود؟
(الف) هویه هوای گرم
(ب) دستگاه Pick And Place
(ج) کوره لحیم کاری
(د) شابلون

۲ در لحیم کاری به کمک هوای گرم از خمیر لحیم استفاده می شود.

صحیح ☐ غلط ☐

۳ سیم لحیم استفاده شده برای مونتاژ قطعات SMD هیچ فرقی با نوع استفاده شده در مونتاژ قطعات مجزا ندارد.

صحیح ☐ غلط ☐

۴ برای پاک کردن روی برد از کدام یک از حلال های زیر استفاده می شود؟

الف) اسپری مخصوص پاک کردن برد
ب) الکل
ج) تینر
د) هر سه مورد

۵ خمیر قلع ترکیبی از و است.

۶ جهت برداشتن قلع اضافی بین پدهای قطعه مونتاژ شده از یا استفاده می شود.

۷ کارگاه هایی که در آن، قسمت عمده ای از عملیات مونتاژ به صورت دستی و برخی به کمک ماشین آلات انجام می شود را کارگاه های گویند.

الف) دستی
ب) تمام اتوماتیک
ج) نیمه اتوماتیک

۸ در شکل ۷۴ ابزار بازرسی چشمی برای قطعات TH را مشاهده می کنید، آیا بزرگ نمایی ذره بین این ابزار برای بازرسی چشمی قطعات SMD مناسب است؟ شرح دهید.

۹ ابزارها و دستگاه های نشان داده شده در شکل ۷۵ چه نام دارند؟ کاربرد آنها را بنویسید.



شکل ۷۵



شکل ۷۴

ارزشیابی شایستگی مونتاژ SMD

نام و نام خانوادگی	شماره ملی	تاریخ ارزشیابی:	نوبت:
کد حرفه	۸۲۱۲۱۲	حرفه :	مونتاژکاران تجهیزات و وسایل برقی و الکترونیکی
کد وظیفه شغل	۸۲۱۲۱۲۰۳	وظیفه شغل:	ساخت مدارهای دیجیتال و میکروکنترلر
کد کار	۸۲۱۲۱۲۰۳۱۲	کار	مونتاژ SMD
	سطح شایستگی	مهارت	۲
	کد پیمان/پودمان	۸۲۱۲۱۲۰۳۱	استاندارد عملکرد: نصب دقیق و باکیفیت قطعات SMD بر روی برد الکترونیکی طبق مشخصات فنی و با رعایت اصول ایمنی.

ردیف	مرا حل کار	شرایط انجام کار (ابزار، مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره دهی)	نتایج ممکن	نمره کسب شده
۱	کنترل سلامت برد و قطعه، انتخاب صحیح قطعه و آماده‌سازی ابزار کالیبره در محیط کار	کارگاه الکترونیک ایمن، زمان‌بندی مشخص، ابزار و مواد سالم و کالیبره، برد و قطعه بررسی شده پیش از شروع	مشاهده مستقیم فرایند آماده‌سازی و بررسی چک‌لیست سلامت تجهیزات و قطعات	رعایت کامل تمیزی برد، سلامت قطعه، کالیبره بودن ابزار	۳	
۲	شناسایی دقیق قطعه، تنظیم ابزار متناسب و اعمال قلع کاری اولیه روی پدها با دقت و مقدار مناسب جهت آماده‌سازی برای نصب نهایی	برد PCB و قطعات SMD، سیم لحیم حاوی روان کننده، هویه با نوک مناسب، ذره‌بین یا لوپ، میز کار تمیز و ایمن و زمان ۱۵-۲۵ دقیقه	بررسی کیفیت و مقدار قلع، نحوه تنظیم هویه و تعداد دفعات قلع کاری	قلع کاری دقیق و یکنواخت با آماده‌سازی کامل پدها	۳	
۳	جاسازی دقیق قطعه، لحیم کاری محکم و بدون عیب و بررسی بصری و فنی کیفیت اتصال و نصب	قطعات الکترونیکی SMD، برد PCB، هویه و سیم لحیم، ابزار SMD و پیچ‌گوشتی، دمنده هوا، محیط ایمن با تهویه مناسب و زمان ۲۵-۳۰ دقیقه	بازرسی با ذره‌بین، تست اتصالات با مولتی‌متر، چک‌لیست کیفیت لحیم کاری و گزارش عملکرد	لحیم کاری بی‌عیب و نصب دقیق قطعات با کنترل کیفیت کامل	۳	
۴	پاکسازی برد از بقایای فلاکس، تکمیل گزارش کار و ارائه توضیحات فنی درباره چگونگی مونتاژ خودکار قطعات SMD	برد مونتاژ شده، محلول پاک‌کننده فلاکس (ایزوپروپیل الکل)، برس نرم یا پد تمیز کننده، دوربین / موبایل، رایانه، محیط کاری تمیز و روشن و زمان ۲۰-۳۰ دقیقه	بررسی تمیزی برد، کامل بودن گزارش و کیفیت تصاویر و مستندات، پرسش درباره مونتاژ SMD	تمیز کاری و مستندسازی دقیق و ارائه کامل و صحیح	۳	
	شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:	کارگاه الکترونیک مجهز به ابزار و تجهیزات مونتاژ، تجهیزات حفاظت فردی و قطعات موردنیاز، دستورالعمل‌های ایمنی و بهداشت حرفه‌ای	مشاهده رفتار هنرجو در حین انجام فعالیت، میزان رعایت اصول ایمنی و نظم، فعالیت‌های گروهی، مدیریت زمان و مسئولیت‌پذیری	رعایت کامل ایمنی، بهداشت و محیط زیست، رفتار حرفه‌ای، نظم و مسئولیت‌پذیری بالا	۴	
				رعایت نسبی ایمنی و بهداشت با خطاهای جزئی، رفتار کاری مناسب	۱	

ارزشیابی کار (شایستگی انجام کار)

☐ بلی☐ خیر

توضیحات:

۱- معیار شایستگی انجام کار:

- کسب حداقل نمره ۲ در مراحل بحرانی شامل شناسایی قطعات، اعمال قلع مناسب، لحیم کاری بدون نقص، کنترل کیفیت، تمیز کاری و مستندسازی قابل قبول است.

- کسب حداقل نمره ۲ از بخش شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش

- کسب حداقل میانگین ۲ از مراحل کار

۲- تعریف سطوح شایستگی: صرف نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

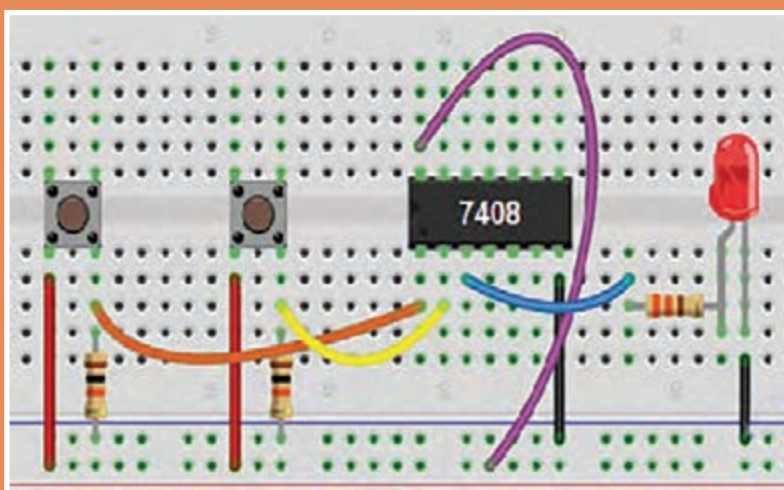
مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها





پودمان دوم

ساخت مدارهای دیجیتالی پایه



واحد یادگیری ۱: اجرای مدار عملگرهای منطقی

آیا تا به حال فکر کرده‌اید

در سامانه‌های دیجیتال، اطلاعات چگونه پردازش و نمایش داده می‌شود؟
سرعت پردازش و انتقال اطلاعات مدارهای دیجیتال بیشتر از سامانه‌های آنالوگ است؟
کدام سامانه اعداد در مدارهای دیجیتال بیشتر استفاده می‌شود؟
سطوح منطقی صفر و یک برای بیان متغیرها چگونه تولید می‌شوند؟
توابع منطقی اصلی در مدارهای دیجیتال چگونه نوشته می‌شوند؟
چه قطعاتی برای ساخت دروازه‌های منطقی به کار می‌رود؟
گیت‌های ترکیبی چگونه ساخته می‌شوند؟
توابع منطقی را به چه دلیل و چگونه ساده می‌کنند؟
چند روش برای ساده‌سازی توابع منطقی وجود دارد؟
چگونه می‌توان تابع منطقی را ساده کرد و آن را نمایش داد؟
یک ماشین حساب گویاترین نمونه سامانه دیجیتال است، نمونه‌های متعدد دیگری از جمله ساعت‌های دیجیتال، نمایشگر چراغ‌های راهنمایی و دستگاه‌های صوتی و تصویری و نوع پیشرفته دیگری مانند رایانه نیز وجود دارند که با یک منطق خاص کار می‌کنند. این منطق معمولاً طبق منطق بولی است که به‌طور مشترک (Logical gate) در تمام مدارهای دیجیتالی یافت می‌شود. در سامانه‌های دیجیتال، یک یا چند دروازه منطقی روی یک یا دو ورودی منطقی عملیات منطقی انجام می‌دهند و در نهایت یک خروجی منطقی را تولید می‌کنند. گیت‌های منطقی از مبنای دودویی یا همان اعداد باینری پیروی می‌کنند. در منطق الکترونیک، هر سطح منطقی نماینده ولتاژ معینی است که این ولتاژ به نوع منطقی که استفاده می‌شود بستگی دارد.

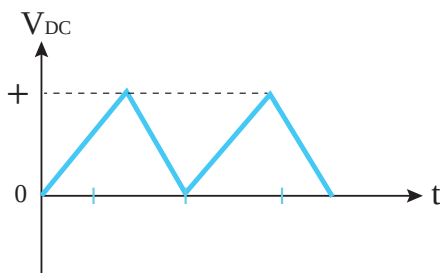
استاندارد عملکرد

ترکیب سیستم‌های اعداد، منطق بولی، کار با گیت‌های منطقی و آی‌سی‌ها برای اجرای مدارهای منطقی ساده

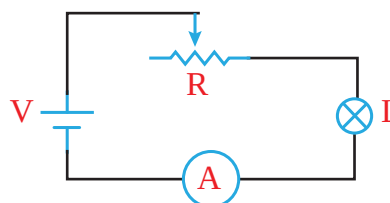
سامانه‌های آنالوگ و دیجیتال

به‌طور کلی همه وسایل الکتریکی و الکترونیکی در قالب سامانه‌های آنالوگ، دیجیتال و یا تلفیقی از آنها ساخته می‌شوند.

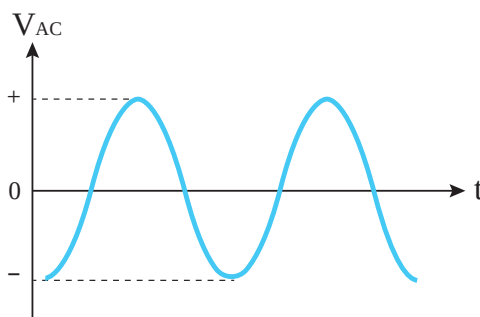
■ **سامانه آنالوگ:** سامانه آنالوگ به سامانه‌ای گفته می‌شود که تغییرات سیگنال در آن به‌صورت پیوسته و منسجم هستند، به عبارت دیگر ولتاژ و جریان آنها در یک بازه زمانی از مقدار کم تا مقدار حداکثر به‌صورت نامساوی یا مساوی تغییر می‌کند یعنی سیگنال و اطلاعات در تمامی لحظات وجود دارد. به دستگاه‌هایی که با چنین سیگنال‌هایی کار می‌کنند دستگاه‌های آنالوگ و به این سامانه‌ها نیز «سامانه‌های آنالوگ» می‌گویند. از جمله دستگاه‌هایی که با این سامانه آنالوگ کار می‌کنند می‌توان به دستگاه کنترل دما و کنترل سرعت با نشان‌دهنده عقربه‌ای اشاره کرد. در شکل ۱ یک نمونه مدار و دو نوع سیگنال آنالوگ نشان داده شده است.



(ب) ولتاژ DC متغیر



(الف) شکل مدار آنالوگ



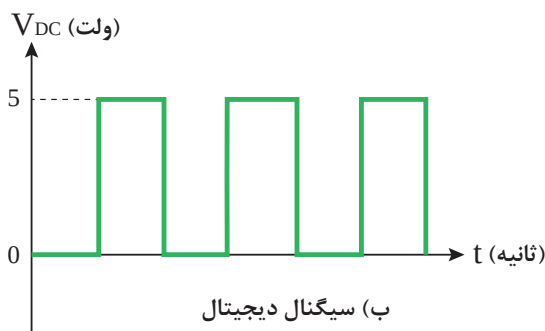
(ج) ولتاژ AC متغیر

شکل ۱

با اتصال مداری مطابق شکل ۱-الف و ایجاد تغییر در مقدار مقاومت متغیر از حداقل تا حداکثر، نور لامپ به آرامی کم و زیاد می‌شود. همچنین با اندازه‌گیری جریان نیز مشاهده می‌کنیم آمپرتر مقدارهای افزایشی و یا کاهش پیوسته را نشان می‌دهد و می‌توان علت پیوسته بودن روشنایی لامپ را به‌خوبی دریافت نمود.

■ **سامانه دیجیتال:** گروه دیگری از سیگنال‌ها وجود دارند که دارای تغییرات پله‌ای و مجزا هستند. به عبارت دیگر ولتاژ و جریان آنها بین دو مقدار حداقل و حداکثر تغییر می‌کند. به دستگاه‌هایی که با چنین سیگنال‌هایی کار می‌کنند دستگاه‌های دیجیتالی و به این سامانه‌ها نیز «سامانه‌های دیجیتال» می‌گویند.

از جمله این دستگاه‌ها می‌توان به ماشین حساب، رایانه و مرکز تلفن اشاره کرد. در شکل ۲ یک نمونه مدار و سیگنال دیجیتالی نشان داده شده است.



شکل ۲

به کمک مدار داده شده در شکل ۲- الف و با قطع و وصل پی در پی کلید مشاهده نور لامپ یا اندازه‌گیری جریان آن متوجه می‌شویم که یک سیگنال دیجیتالی به وجود می‌آید و هنگام قطع و وصل کلید، لامپ دو حالت روشن و خاموش را به خود می‌گیرد.

در منزل، هنرستان یا محیط کار دستگاه‌هایی را بیابید که فقط دارای سامانه آنالوگ، فقط دارای سامانه دیجیتال یا دارای سامانه تلفیقی آنالوگ و دیجیتال هستند. نتیجه را در قالب یک گزارش ارائه دهید.

جستجو کنید



سیستم‌های اعداد

■ **سیستم دهدهی (دسیمال - Decimal):** سیستمی که در زندگی روزمره با آن سروکار داریم سیستم اعداد دهدهی یا اعشاری است. این سیستم اعداد برای ماشین‌های محاسبه‌گر الکترونیکی سیستم مناسبی نیست، زیرا در این مدارها تفکیک ده حالت از یکدیگر میسر نیست و میزان اشتباه و خطا را افزایش می‌دهد. در سیستم دهدهی هر رقم دارای یک ارزش مکانی است، مثلاً در عدد ۹۸۷۶۵ می‌توان ارزش هر یک از رقم‌ها را به صورت عبارت نشان داده شده هم نوشت.

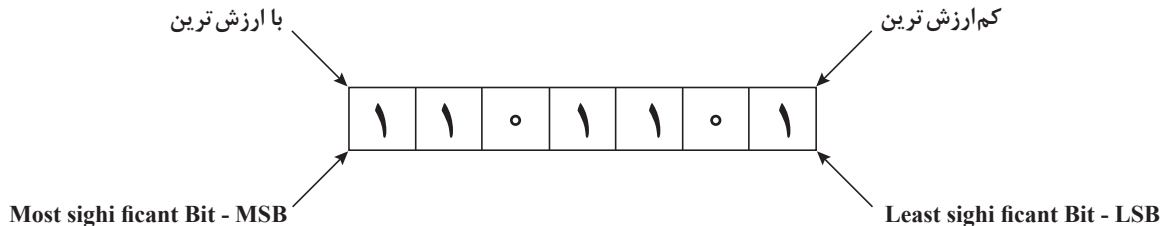
$$98765 = 9 \times 10000 + 8 \times 1000 + 7 \times 100 + 6 \times 10 + 5 \times 1$$

$$= 9 \times 10^4 + 8 \times 10^3 + 7 \times 10^2 + 6 \times 10^1 + 5 \times 10^0$$

در واقع رقم ۹ پر ارزش‌ترین و رقم ۵ کم ارزش‌ترین رقم عدد ۹۸۷۶۵ هستند.

■ **سیستم دودویی (باینری - Binary):** چون عناصر نیمه هادی مانند دیود و ترانزیستور اجزای دو حالتی هستند، یعنی می‌توانند دو حالت هدایت جریان (بستن کلید) یا حالت قطع جریان (باز کردن کلید) را داشته باشند. اگر هروضعیت این اجزا را معادل یک علامت در نظر بگیریم، تنها سیستم اعداد مناسب برای این ماشین‌ها، سیستمی با دو رقم خواهد بود. در سیستم اعداد دودویی، از دو نشانه یا رقم (۰) و (۱) استفاده می‌شود. به این سیستم اعداد، سیستم اعداد دودویی یا در پایه «۲» نیز گفته می‌شود. شمارش در پایه «۲» مشابه شمارش در پایه «۱۰» است.

برای مثال، در عدد دودویی ۱۱۰۱۱۰۱ رقم سمت راست کمترین ارزش مکانی و رقم سمت چپ دارای بیشترین ارزش مکانی است. در این مثال کم ارزش ترین بیت دارای ارزشی برابر با $2^0 = 1$ است که به آن LSB گفته می شود و با ارزش ترین بیت دارای ارزشی برابر $2^6 = 64$ است که به آن MSB گفته می شود (شکل ۳).



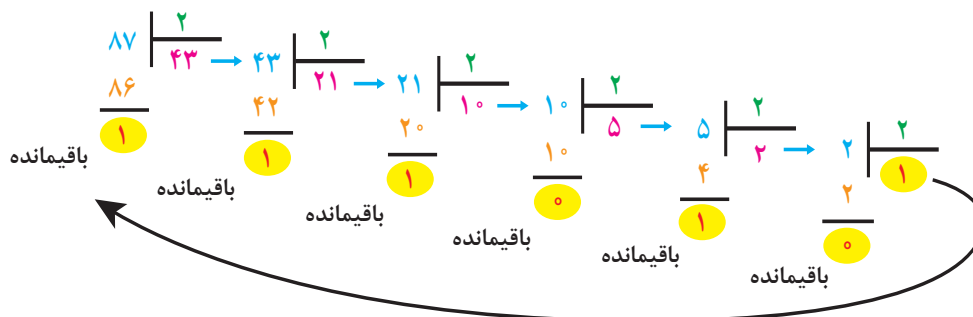
شکل ۳

ارزش مکانی سایر ارقام این عدد نیز در عبارت زیر مشخص شده است.

$$1101101 = 1 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

چون ما در کارهای روزمره و مدارها با سیستم های ددهی و باینری سروکار داریم لذا می بایست با چگونگی تبدیل آنها آشنا شویم.

■ **تبدیل اعداد ددهی (دسیمال - Decimal) به اعداد دودویی (باینری - Binary):** برای تبدیل اعداد ددهی به اعداد دودویی، از روش تقسیم های متوالی عدد ددهی بر ۲ استفاده می کنیم. به مثال شکل ۴ توجه کنید.



از آخرین خارج قسمت به ترتیب باقیمانده ها را می خوانیم

شکل ۴- تبدیل اعداد با مبنای ۱۰ به ۲ با روش تقسیم متوالی

■ **تبدیل اعداد دودویی (باینری - Binary) به اعداد ددهی (دسیمال - Decimal):**

برای به دست آوردن معادل ددهی هر عدد باینری، می توان از روش ارزش مکانی استفاده کرد. برای این منظور کافی است ارزش مکانی هر رقم را نوشته و ارزش مکانی ارقام آن عدد را با هم جمع کنیم. به این ترتیب معادل ددهی (اعشاری) به دست می آید.



معادل دهدهی (دسیمال) عدد دودویی (باینری) ۱۱۰۱۱۰۱ برابر است با:

$$\begin{aligned}(1101101)_2 &= (1 \times 2^6) + (1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) \\ &= 64 + 32 + 0 + 8 + 4 + 0 + 1 \\ &= (109)_{10}\end{aligned}$$

یا به اختصار می‌توان نوشت:

$$\begin{array}{ccccccc} 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 64 & 32 & 16 & 8 & 4 & 2 & 1 \end{array} \quad 1 = 64 + 32 + 8 + 4 + 1 = (109)_{10}$$

در سیستم‌های دیجیتال به کوچک‌ترین فضای حافظه که برای ذخیره هر یک از ارقام سیستم باینری استفاده می‌شود، «بیت» (Bit) گفته می‌شود. همچنین به هر ۸ بیت یک «بایت» (Byte) و به یک یا چند بایت «کلمه» یا «واژه» (word) گفته می‌شود.



عدد $(100011111101101)_2$ دارای بیت و بایت است.

امروزه در سامانه‌های دیجیتال از سیستم‌های اعداد دیگر همچون مبنای هشت و شانزده نیز استفاده می‌شود.

■ مبنای ۸ (اکتال Octal) و مبنای ۱۶ (هگزادسیمال Hexadecimal)

برای مختصرنویسی اعداد باینری معمولاً آنها را در مبنای هشت یا شانزده می‌نویسند. این کار را به دو دلیل انجام می‌دهند؛ نخست آنکه در نوشتن، خواندن و انتقال زنجیره «۰» و «۱» اعداد باینری احتمال خطا زیاد است، لذا بیان اعداد در مبنای دیگر (با مبنای بزرگ‌تر) یک ضرورت است. **دلیل دوم** این است که باید مبنای عددی را طوری انتخاب کنیم تا تبدیل اعداد باینری به این مبنایها و برعکس، خیلی سریع انجام شود. مبنای هشت و مبنای شانزده هر دو دارای این ویژگی هستند چون می‌توان آنها را در مبنای ۲ نوشت:

$$(2^3 = 8 \text{ و } 2^4 = 16)$$

همان‌طوری که در سیستم دهدهی از ۱۰ نشانه یا رقم و در سیستم دودویی از ۲ نشانه یا رقم استفاده می‌شود در سیستم ۸ (اکتال) و سیستم ۱۶ (هگزادسیمال) نیز به ترتیب از هشت و شانزده نشانه یا رقم استفاده می‌شود. در جدول ۱ ارقام یا نشانه‌های مربوط به هر یک از سیستم‌ها (مبنایها) آورده شده است.

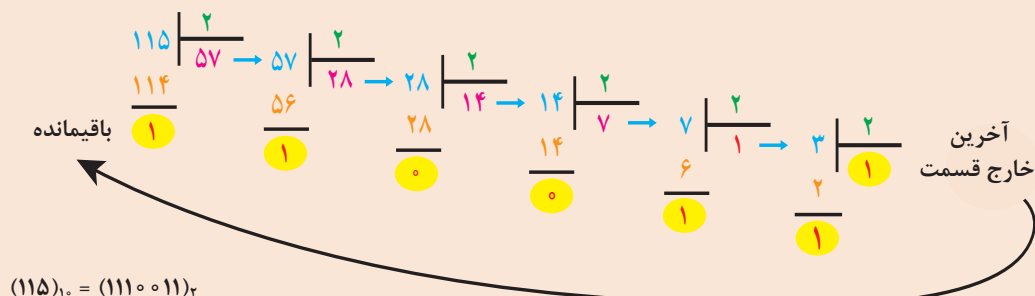
جدول ۱

Numbering System		
System	Base	Digits
Binary	۲	۰ ۱
Octal	۸	۰ ۱ ۲ ۳ ۴ ۵ ۶ ۷
Decimal	۱۰	۰ ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹
Hexadecimal	۱۶	۰ ۱ ۲ ۳ ۴ ۵ ۶ ۷ ۸ ۹ A B C D E F

مثال

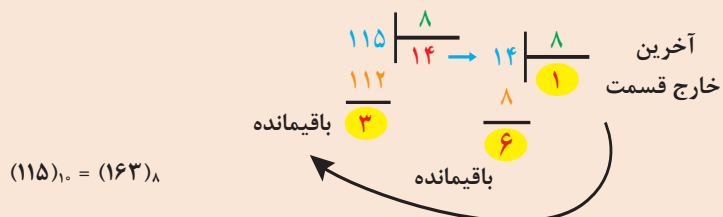


عدد دهدهی ۱۱۵ را در مبنای ۲ و مبنای ۸ و عدد ۳۳ را در مبنای ۱۶ بنویسید.
حل: الف) تبدیل اعداد دهدهی به دودویی (مبنای ۱۰ به مبنای ۲) در شکل ۵ آمده است.



شکل ۵- تبدیل اعداد با مبنای ۱۰ به ۲ با روش تقسیم متوالی

ب) تبدیل اعداد دهدهی به اُکتال (Octal) (مبنای ۱۰ به مبنای ۸) در شکل ۶ آمده است.



شکل ۶- تبدیل اعداد با مبنای ۱۰ به ۸ با روش تقسیم متوالی

ج) تبدیل اعداد دهدهی به هگزا دسیمال (Hex) (مبنای ۱۰ به مبنای ۱۶) در شکل ۷ آمده است.



شکل ۷- تبدیل اعداد با مبنای ۱۰ به ۱۶ با روش تقسیم متوالی

چون در سیستم‌های دیجیتال ما با اعداد ۰ تا ۱۵ سرو کار داریم لذا در جدول ۲ اعداد دهدهی ۰ تا ۱۵ را در مبنای ۲، مبنای ۸ و مبنای ۱۶ مشاهده می‌کنید.

جدول ۲

مبنای ۱۰ (دسیمال)	مبنای ۲ (باینری)	مبنای ۸ (اکتال)	مبنای ۱۶ (هگزادسیمال)
۰	۰	۰	۰
۱	۱	۱	۱
۲	۱۰	۱۰	۱۰
۳	۱۱	۱۱	۱۱
۴	۱۰۰	۱۰۰	۱۰۰
۵	۱۰۱	۱۰۱	۱۰۱
۶	۱۱۰	۱۱۰	۱۱۰
۷	۱۱۱	۱۱۱	۱۱۱
۸	۱۰۰۰	—	۱۰۰۰
۹	۱۰۰۱	—	۱۰۰۱
۱۰	۱۰۱۰	—	A
۱۱	۱۰۱۱	—	B
۱۲	۱۱۰۰	—	C
۱۳	۱۱۰۱	—	D
۱۴	۱۱۱۰	—	E
۱۵	۱۱۱۱	—	F

■ تبدیل اعداد در سیستم‌های عددنویسی مختلف به یکدیگر:

در روش عمومی برای تبدیل اعداد در مبنای غیر از ۱۰، مانند تبدیل (۲ به ۸)، (۲ به ۱۶)، (۸ به ۱۶) و یا بالعکس ابتدا باید عدد مورد نظر را به مبنای ۱۰ ببریم، سپس با تقسیم‌های متوالی بر مبنای عدد خواسته شده نتیجه را به دست آوریم. اما روش ساده‌تری نیز برای تبدیل این مبناها به یکدیگر وجود دارد که سرعت کار را بالاتر می‌برد. جدول ۲، شکل نمایش اعداد ۰ تا ۱۵ را در مبناهای مختلف اما به صورت باینری نشان می‌دهد. با به کارگیری این روش و از طریق تمرین زیاد به راحتی می‌توانید تبدیل هر عدد در مبناهای مختلف را بدون استفاده از محاسبات و خیلی ساده و سریع به دست آورید. در ادامه به تشریح این روش می‌پردازیم.

■ **تبدیل مبنای ۸ به ۲ و بالعکس:** چون بزرگ‌ترین عدد مبنای اکتال ۷ است که معادل آن در شکل باینری برابر (۱ ۱ ۱) است لذا برای تبدیل یک عدد اکتال به عدد باینری معادل آن، باید از سمت راست به جای هر رقم عدد اکتال، معادل باینری آن را به صورت سه رقم بنویسیم.

عدد ۳۲۶۵ در مبنای ۸ را به عدد باینری تبدیل کنید.

$$(3265)_8 = (\underbrace{011}_3 \underbrace{0101}_2 \underbrace{0101}_6 \underbrace{0101}_5)_2$$

مطابق جدول برای هر رقم اکتال سه رقم باینری معادل آن را قرار می‌دهیم

مثال



برای تبدیل اعداد مبنای ۲ به ۸ نیز کافی است از سمت راست عدد باینری سه رقم سه رقم جدا کرده و به جای هر سه رقم باینری معادل اکتال آن را قرار داد.

مثال



معادل عدد $(1010111001110011)_2$ در مبنای ۸ را به دست آورید.

$$\begin{aligned} (1010111001110011)_2 &= (?)_8 \\ \begin{array}{cccccc} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 1 & 2 & 7 & 1 & 6 & 3 \end{array} \\ (1010111001110011)_2 &= (127163)_8 \end{aligned}$$

■ **تبدیل مبنای ۱۶ به ۲ و بالعکس:** چون بزرگ‌ترین عدد مبنای هگزادسیمال عدد ۱۵ است که معادل آن در شکل باینری برابر (1111) است لذا برای تبدیل یک عدد هگزادسیمال به عدد باینری معادل آن باید از سمت راست به جای هر رقم عدد هگزادسیمال معادل باینری آن را به صورت چهار رقم باینری بنویسیم، به مثال داده شود توجه نمایید تا با چگونگی تبدیل آشنا شوید.

مثال



عدد $D5F9$ در مبنای ۱۶ را به عدد باینری تبدیل کنید.

$$(D5F9)_{\text{HEX}} = (?)_2$$

با توجه به جدول ۳ ارزش هر رقم هگزا دسیمال در شکل باینری به صورت زیر است.

جدول ۳

D	۵	F	۹																
$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$																
<table border="1"><tr><td>۱</td><td>۱</td><td>۰</td><td>۱</td></tr></table>	۱	۱	۰	۱	<table border="1"><tr><td>۰</td><td>۱</td><td>۰</td><td>۱</td></tr></table>	۰	۱	۰	۱	<table border="1"><tr><td>۱</td><td>۱</td><td>۱</td><td>۱</td></tr></table>	۱	۱	۱	۱	<table border="1"><tr><td>۱</td><td>۰</td><td>۰</td><td>۱</td></tr></table>	۱	۰	۰	۱
۱	۱	۰	۱																
۰	۱	۰	۱																
۱	۱	۱	۱																
۱	۰	۰	۱																

در نتیجه شکل باینری عدد چنین خواهد شد

$$(D5F9)_{\text{HEX}} = (110101011111001)_2$$

برای تبدیل اعداد مبنای ۲ به ۱۶ نیز کافی است از سمت راست عدد باینری چهاررقم چهاررقم جدا کرده و به جای هر چهار رقم باینری معادل هگزا دسیمال آن را قرار داد.

مثال



معادل عدد $(1010111001110011)_2$ در مبنای ۱۶ را به دست آورید.

حل: براساس دسته‌بندی چهارتایی انجام شده در ارقام عدد باینری و استفاده از جدول ۳ می‌توان تبدیل را به صورت زیر انجام داد.

$$\begin{aligned} (1010111001110011)_2 &= (?)_{\text{HEX}} \\ \begin{array}{cccc} \downarrow & \downarrow & \downarrow & \downarrow \\ 1010 & 1110 & 0111 & 0011 \\ \downarrow & \downarrow & \downarrow & \downarrow \\ A & E & 7 & 3 \end{array} \\ (1010111001110011)_2 &= (AE73)_{\text{HEX}} \end{aligned}$$

■ **تبدیل مبنای ۸ به ۱۶ و بالعکس:** ساده‌ترین راه برای تبدیل اعداد مبنای ۸ به مبنای ۱۶ و یا بالعکس می‌بایست ابتدا عدد موردنظر در هر یک از مبناهای ۸ یا ۱۶ را به مبنای ۲ برده و سپس براساس روش‌هایی که در قسمت‌های قبل توضیح داده شده معادل آن را به‌دست آورد.

اعداد داده شده را در مبناهای خواسته شده به‌دست آورید.

$$(۵۳۷)_8 = (?)_{16}$$

$$(۹۲۶)_{16} = (?)_8$$

کار در کلاس



نکته مهم



امروزه اغلب ماشین حساب‌ها دارای انواع تبدیل‌های ذکر شده هستند و عملاً نیازی به محاسبه این تبدیل‌ها وجود ندارد. آموزش داده شده به منظور درک عمیق‌تر مفاهیم مرتبط با دیجیتال و میکروکنترلر است.

با مراجعه به رسانه‌های مختلف، حداقل دو نمونه مبدل را بیابید و با آن تمرین کنید. نتیجه را در قالب گزارش ارائه دهید.

پژوهش



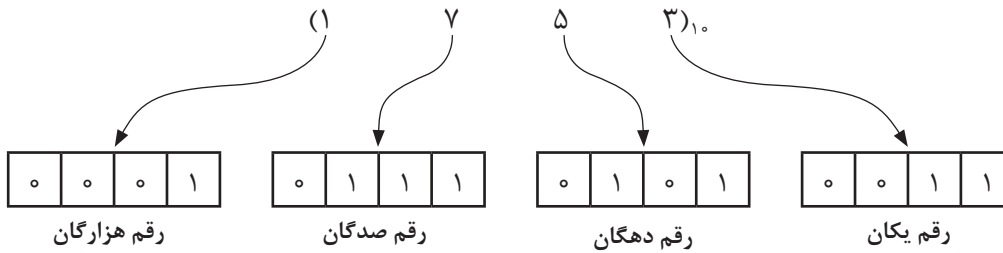
کد BCD

جدول ۴

مبنا ۱۰ (دسیمال)	مبنا ۲ (باینری)	کد BCD
۰	۰	۰۰۰۰
۱	۱	۰۰۰۱
۲	۱۰	۰۰۱۰
۳	۱۱	۰۰۱۱
۴	۱۰۰	۰۱۰۰
۵	۱۰۱	۰۱۰۱
۶	۱۱۰	۰۱۱۰
۷	۱۱۱	۰۱۱۱
۸	۱۰۰۰	۱۰۰۰
۹	۱۰۰۱	۱۰۰۱

بعضی از ماشین‌های محاسبه‌گر الکترونیکی، عملیات ریاضی را در (سیستم - مبنا) کد BCD (Binary Coded Decimal) انجام می‌دهند. در کد BCD هر رقم دهدهی را با چهاررقم باینری معادل آن نشان می‌دهند. تبدیل اعداد دهدهی به معادل BCD آنها به مراتب از تبدیل اعداد دهدهی به معادل باینری آنها ساده‌تر است، زیرا برای این تبدیل، دانستن معادل باینری ارقام صفر تا ۹ کفایت می‌کند. توجه داشته باشید که در این روش نمایش اعداد، باید هر رقم دهدهی را با چهار رقم (بیت) باینری نمایش دهیم. در جدول ۴ تفاوت نمایش ارقام دهدهی صفر تا ۹ به‌صورت باینری و BCD نشان داده شده است.

پس بر اساس توضیحات و جدول ۴ معادل BCD عدد ددهی (۱۷۵۳) را مطابق شکل ۸ می توان نشان داد.



$$(1753)_{10} = (1011101010011)_{BCD}$$

شکل ۸- معادل BCD عدد مبنای دسیمال

اعداد داده شده زیر را در مبناهای خواسته شده به دست آورید.

- الف) $(125)_{10} = (?)_2 = (?)_{BCD} = (?)_8 = (?)_{16}$
 ب) $(432)_8 = (?)_{10} = (?)_{16} = (?)_{BCD} = (?)_2$
 ج) $(10101010)_2 = (?)_8 = (?)_{16} = (?)_{10} = (?)_{BCD}$
 د) $(968)_{BCD} = (?)_{16} = (?)_8 = (?)_2 = (?)_{10}$
 هـ) $(2C7F)_{16} = (?)_2 = (?)_{10} = (?)_{BCD} = (?)_8$

کار در منزل



سیگنال آنالوگ و دیجیتال و سیستم اعداد

هدف: کسب مهارت در تشخیص سیگنال‌های آنالوگ، دیجیتال و تبدیل سیستم‌های اعداد به یکدیگر
مواد، ابزار و تجهیزات: لوازم التحریر - اینترنت - رایانه - نرم افزار مولتی سیم یا هر نرم افزار مناسب دیگر -
 نرم افزار PC Binary Converter یا نرم افزار مشابه - دستگاه ماشین حساب یا ماشین حساب Windows
 رایانه

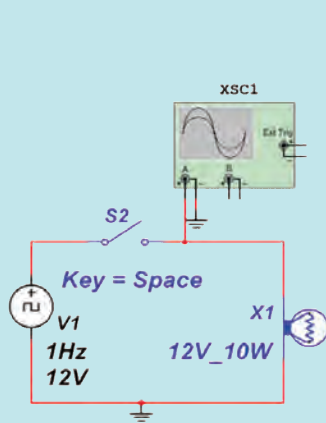
مراحل اجرای کار:

- ۱ مدار شکل ۹ را در فضای نرم افزاری ببندید.
- ۲ منبع ورودی را سیگنال مربعی با دامنه ۱۲ ولت و فرکانس ۵۰ هرتز انتخاب کنید. سپس کلید S_1 را وصل کنید.
- ۳ شکل موج خروجی دو سر خازن را روی صفحه اسیلوسکوپ نرم افزار به صورت پایدار ظاهر کنید.
- ۴ آیا با مشاهده شکل موج خروجی می توان نتیجه گرفت که سیگنال شارژ و دشارژ خازن، یک نوع سیگنال آنالوگ است؟ بلی ☐ خیر ☐

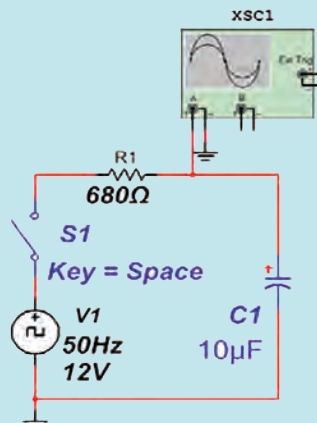
کار عملی ۱



۵ مدار شکل ۱۰ را در محیط نرم‌افزاری ببندید. سپس کلید S_p را وصل کنید.



شکل ۱۰



شکل ۹

۶ شکل موج خروجی را روی صفحه اسیلوسکوپ نرم افزار به صورت پایدار ظاهر کنید.

۷ آیا با مشاهده شکل موج خروجی می‌توان نتیجه گرفت که سیگنال دو سر لامپ یک سیگنال دیجیتال است؟ بلی ☐ خیر ☐

۸ مداری رسم کنید که بتواند باز و بسته بودن یک در را به سطوح منطقی تبدیل کند.

۹ اعداد جدول ۵ را که در سیستم دهدهی (Decimal) هستند، به سیستم‌های باینری، اُکتال و هگزادسیمال تبدیل کنید.

۱۰ با استفاده از نرم‌افزار PC Binary Converter یا نرم‌افزارهای مشابه و دستگاه ماشین حساب، شکل ۱۱ (ماشین حساب Windows) تبدیل مبنای اعداد جدول ۵ را راستی‌آزمایی کنید.

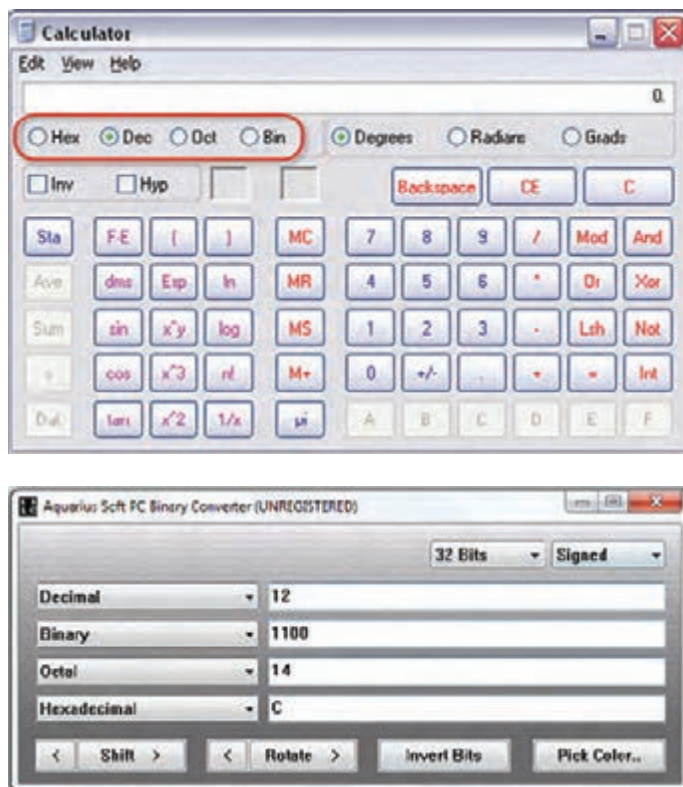
جدول ۵

مبنای ۱۰ (Decimal)	مبنای ۲ (Binary)	مبنای ۸ (Octal)	مبنای ۱۶ (Hexadecimal)
۵			
۱۲			
۱۷			
۳۶			
۷۵			
۱۲۸			
۱۸۰			
۴۸۰			
۵۱۲			

نکته



نرم افزار PC Binary Converter مبدل سامانه های اعداد است. این نرم افزار رایگان و قابل بارگیری از اینترنت است. قابلیت های دیگر این نرم افزار را مشخص کنید و نتایج را به طور خلاصه بنویسید.



شکل ۱۱- نرم افزار مبدل و ماشین حساب

اعداد زیر را به صورت کد BCD بنویسید.

الف) ۵۷
ب) ۷۸۶
ج) ۹۴۳
د) ۲۹۱

فعالیت کلاسی



سطوح منطقی صفر و یک

همان گونه که اشاره شد در مدارهای دیجیتالی دو حالت وجود دارد. این حالت ها را با اصطلاحات مختلفی بیان می کنند، مثلاً برای نشان دادن حالت روشن و حالت خاموش لامپ، از اصطلاحات مختلفی استفاده می شود که در جدول ۶ نشان داده شده است.

جدول ۶

صفر	وجود نداشتن	low	OFF	کلید قطع	لامپ خاموش
یک	وجود داشتن	high	ON	کلید وصل	لامپ روشن

هر یک از این اصطلاحات بیانگر مفهوم خاصی از مدار است، برای مثال «ON - OFF» روشن و خاموش بودن، «low - high» حداکثر و حداقل بودن ولتاژ اعمال شده به مدار و «۱ - ۰» معرف بودن و نبودن است. معمولاً برای خلاصه‌نویسی بیان مفاهیم و عبارات دیجیتالی بیشتر از صفر و یک استفاده می‌شود. البته باید دقت داشت که صفر و یک به کار رفته در مدارهای دیجیتالی با صفر و یک عبارات جبری یکسان نیست. در مباحث دیجیتال ۰ و ۱ نشان‌دهنده وضعیت مدار هستند در صورتی که در ریاضی مفهوم صفر و یک بیانگر مقدار عددی است.

سطح ولتاژ در مدارهای دیجیتالی مقادیری در محدوده (۰ تا ۵ ولت) یا (۰ تا ۱۵ ولت) است. مقادیر ۵ و ۱۵ ولت سطوح ولتاژ ماکزیمم (یک منطقی) و سطح ولتاژ مینیمم (حدود صفر ولت) بیانگر (صفر منطقی) است.

■ دروازه‌های منطقی پایه

دروازه‌های منطقی (گیت‌ها - GATES) مطابق شکل ۱۲ عملگرهایی با یک یا چند ورودی و یک خروجی هستند. برای ترسیم دروازه‌های منطقی از علامت‌های اختصاری استفاده می‌کنیم.

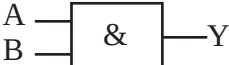


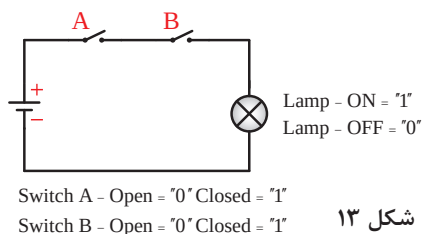
شکل ۱۲

هر دروازه منطقی دارای تعریف، علامت اختصاری، تابع منطقی، جدول صحت (جدول درستی) و منطق کاری خاص خود هست که در ادامه با هریک از این موارد آشنا خواهید شد.

■ **دروازه AND:** به گیت یا عملگری گفته می‌شود که خروجی آن زمانی وجود دارد که همه ورودی‌های آن وجود داشته باشند. سایر مشخصات گیت AND در جدول ۷ نشان داده شده است.

جدول ۷

علامت اختصاری		تابع منطقی	جدول صحت (درستی)															
ANSI	IEC																	
		$Y = A \cdot B$	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	0	1	0	0	1	1	1
A	B	Y																
0	0	0																
0	1	0																
1	0	0																
1	1	1																



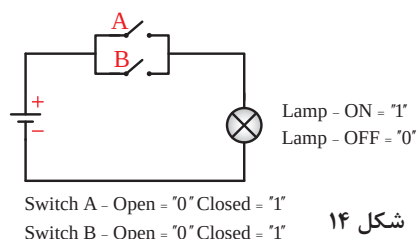
شکل ۱۳

شکل ۱۳ مدار معادل کلیدی گیت AND را نشان می‌دهد. با قطع و وصل کلیدهای A و B می‌توان با چگونگی عملکرد آن آشنا شده و به نتایجی مشابه جدول صحت دست یافت.

■ **دروازه OR:** به گیت یا عملگری گفته می‌شود که خروجی آن زمانی وجود دارد که حداقل یکی از ورودی‌های آن وجود داشته باشند. سایر مشخصات گیت OR در جدول ۸ نشان داده شده است.

جدول ۸

علامت اختصاری		تابع منطقی	جدول صحت (درستی)															
ANSI	IEC																	
		$Y = A + B$	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	1
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	1																

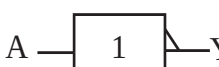


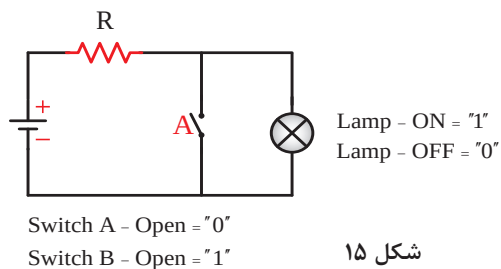
شکل ۱۴

شکل ۱۴ مدار معادل کلیدی گیت OR را نشان می‌دهد. با قطع و وصل کلیدهای A و B با چگونگی عملکرد آن آشنا شده و به نتایجی مشابه جدول صحت دست خواهید یافت.

■ **دروازه NOT:** به گیت یا عملگری گفته می‌شود که خروجی آن زمانی وجود دارد که ورودی‌های آن وجود نداشته باشند. سایر مشخصات گیت NOT در جدول ۹ نشان داده شده است.

جدول ۹

علامت اختصاری		تابع منطقی	جدول صحت (درستی)						
ANSI	IEC								
		$Y = \overline{A} = A'$	<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table>	A	Y	0	1	1	0
A	Y								
0	1								
1	0								



شکل ۱۵

شکل ۱۵ مدار معادل کلیدی گیت AND را نشان می‌دهد. با قطع و وصل کلید A می‌توان با چگونگی عملکرد آن آشنا شده و به نتایجی مشابه جدول صحت دست یافت (شکل ۱۵).

با بررسی مدار معادل کلیدی گیت‌های AND، OR و NOT عملکرد هر یک را با جداول صحت مقایسه کنید.

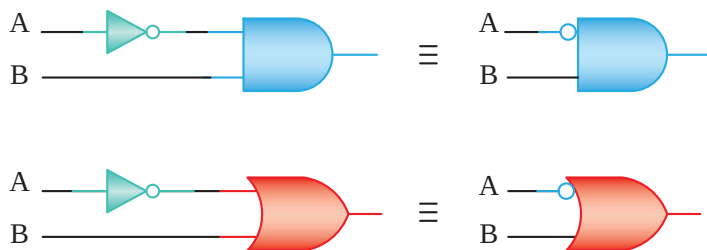
فعالیت کلاسی



نکته



اگر گیت NOT در ابتدای هر دروازه‌ای قرار گیرد به جهت خلاصه‌نویسی می‌توان به جای رسم گیت NOT از علامت دایره در ورودی آن استفاده کرد. به عنوان مثال تصاویر شکل ۱۶ این مطلب را در دو گیت AND و OR نشان می‌دهد.

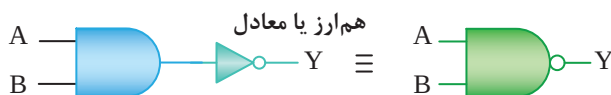


شکل ۱۶

■ دروازه‌های منطقی ترکیبی

با ترکیب برخی دروازه‌های منطقی پایه با یکدیگر، دروازه‌های منطقی جدیدی ساخته می‌شوند که در مدارهای دیجیتال کاربرد دارند.

■ **دروازه NAND – «نفی AND»:** به گیت یا عملگری گفته می‌شود که خروجی آن زمانی وجود دارد که حداقل یکی از ورودی‌های آن وجود نداشته باشند. در واقع خروجی آن نفی دروازه AND است. در دروازه‌های منطقی NAND ابتدا ورودی‌ها AND شده و سپس NOT می‌شوند (شکل ۱۷).



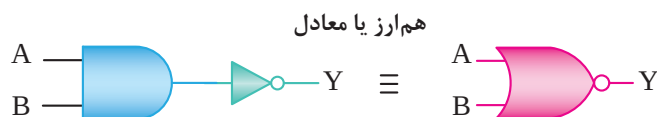
شکل ۱۷

سایر مشخصات گیت NAND در جدول ۱۰ نشان داده شده است.

جدول ۱۰

علامت اختصاری		تابع منطقی	جدول صحت (درستی)															
ANSI	IEC																	
		$Y = \overline{A \cdot B} = (AB)'$	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	1	0	1	1	1	0	1	1	1	0
A	B	Y																
0	0	1																
0	1	1																
1	0	1																
1	1	0																

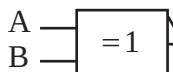
■ **دروازه NOR – «نفی OR»:** به گیت یا عملگری گفته می‌شود که خروجی آن زمانی وجود دارد که همه ورودی‌های آن وجود نداشته باشند. در واقع خروجی آن نفی دروازه OR است. در دروازه‌های منطقی NOR ابتدا ورودی‌ها OR شده و سپس NOT می‌شوند (شکل ۱۸).



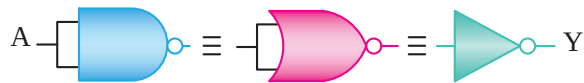
شکل ۱۸

سایر مشخصات گیت NOR در جدول ۱۱ نشان داده شده است

جدول ۱۱

علامت اختصاری		تابع منطقی	جدول صحت (درستی)															
ANSI	IEC																	
		$Y = \overline{A + B} = (A + B)'$	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

در صورت عدم دسترسی به گیت NOT از گیت‌های NAND و NOR مشابه شکل ۱۹ می‌توان به‌عنوان گیت NOT استفاده کرد.



شکل ۱۹

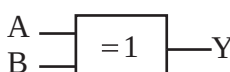
بررسی کنید خروجی توابع منطقی هر یک از گیت‌های شکل ۱۹ با گیت NOT برابر است.

فعالیت کلاسی

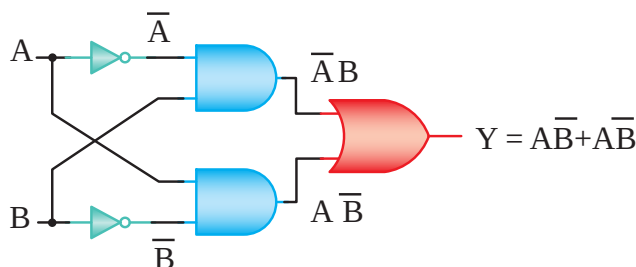


■ **دروازه OR انحصاری (Exclusive OR – XOR):** این دروازه دارای دو ورودی و یک خروجی است. XOR گیت یا عملگری است که خروجی آن زمانی وجود دارد که دو ورودی آن برابر نباشند، به عبارت دیگر دو ورودی در سطح منطقی مشابه نباشند (دو ورودی مخالف یکدیگر باشند). سایر مشخصات گیت XOR در جدول ۱۲ نشان داده شده است.

جدول ۱۲

علامت اختصاری		تابع منطقی	جدول صحت (درستی)															
ANSI	IEC																	
		$Y = \overline{\overline{A} + \overline{B}} = (A + B)$	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table>	A	B	Y	0	0	0	0	1	1	1	0	1	1	1	0
A	B	Y																
0	0	0																
0	1	1																
1	0	1																
1	1	0																

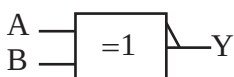
با توجه به تابع منطقی گیت XOR می‌توان مدار معادل منطقی را با استفاده از دروازه‌های منطقی پایه به صورت شکل ۲۰ رسم کرد.



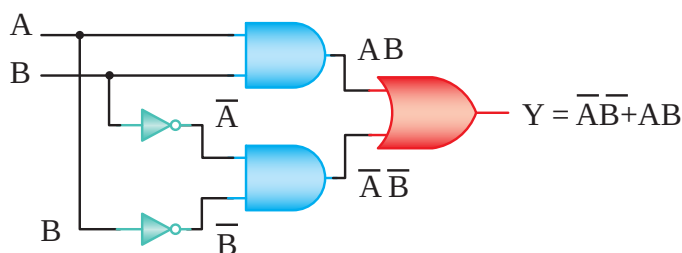
شکل ۲۰- مدار معادل XOR

■ **دروازه منطقی NOR انحصاری (Exclusive NOR – XNOR):** این دروازه هم دارای دو ورودی و یک خروجی است. XNOR گیت یا عملگری است که خروجی آن زمانی وجود دارد که ورودی‌های آن برابر باشند. به عبارت دیگر دو ورودی در سطح منطقی مشابه باشند (دو ورودی یکسان باشند). در واقع خروجی این گیت نفی (NOT) دروازه XOR است. سایر مشخصات گیت XNOR در جدول ۱۳ نشان داده شده است.

جدول ۱۳

علامت اختصاری		تابع منطقی	جدول صحت (درستی)															
ANSI	IEC																	
		$Y = \overline{\overline{A}B} + A\overline{B} = A \odot B$ $Y = A \odot B = \overline{A \oplus B}$	<table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table>	A	B	Y	0	0	1	0	1	0	1	0	0	1	1	1
A	B	Y																
0	0	1																
0	1	0																
1	0	0																
1	1	1																

با توجه به تابع منطقی گیت XNOR می‌توان مدار معادل منطقی را با استفاده از دروازه‌های منطقی پایه به صورت شکل ۲۱ برای آن رسم کرد.



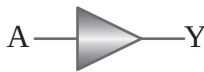
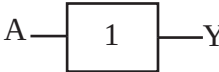
شکل ۲۱- مدار معادل XNOR

■ دروازه منطقی بافر (Buffer):

در برخی مدارهای منطقی از دروازه یا گیت Buffer که عملکرد آن تقویت میزان جریان‌دهی مدار است استفاده می‌شود. این دروازه دارای یک ورودی و یک خروجی است. طبق تعریف «بافر» به گیت یا عملگری گفته می‌شود که خروجی آن زمانی وجود دارد که ورودی آن وجود داشته باشد.

سایر مشخصات گیت Buffer در جدول ۱۴ نشان داده شده است.

جدول ۱۴

علامت اختصاری		تابع منطقی	جدول صحت (درستی)						
ANSI	IEC								
		$Y = A$	<table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	A	Y	0	0	1	1
A	Y								
0	0								
1	1								

مدار معادل کلیدی هریک از گیت‌های NAND، XOR، XNOR و Buffer را رسم کنید

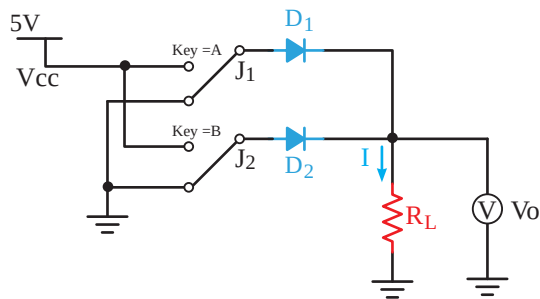
پرسش



تکنولوژی ساخت مدارهای منطقی

در گذشته مفاهیم و عملکرد مدارهای دیجیتالی با استفاده از قطعات پایه و مدارهای الکترونیکی مانند مقاومت، دیود و ترانزیستور تحت عنوان مدارهای RTL و RDL ساخته و استفاده می‌شد.

با پیشرفت علم الکترونیک ساخت این مدارها نیز توسعه چشمگیری یافته است. به‌طوری که امروز از مدارهای



شکل ۲۲

مجتمع (IC) با تکنولوژی‌های ساخت جدیدتر به‌نام‌های TTL، MOS، CMOS و ECL استفاده می‌شود. در اینجا چون هدف پرداختن به جزئیات این تکنولوژی‌ها نیست لذا فقط به چند نمونه مدار RDL و RTL گیت‌های پایه اشاره شده است.

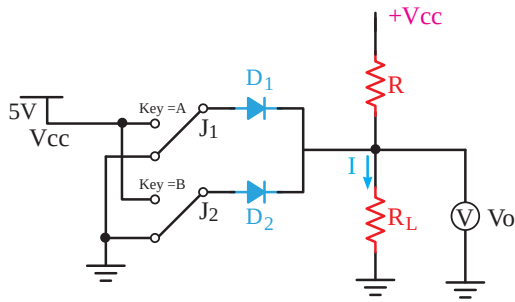
شکل ۲۲ مدار معادل RDL گیت OR دو ورودی را نشان می‌دهد که با استفاده از دو دیود ساخته شده است.

در این مدار، اگر با تغییر وضعیت کلیدهای A و B که در ورودی قرار گرفته‌اند مطابق جدول صحت، (جدول ۱۵) به‌ترتیب به زمین (ولتاژ صفر - L) و ولتاژ (۵ ولت - H) وصل کنیم مشاهده خواهیم کرد که در شرایطی

جدول ۱۵

A	B	D_1	D_2	V_o
L	L	قطع	قطع	L
L	H	قطع	هدایت	H
H	L	هدایت	قطع	H
H	H	هدایت	هدایت	H

که دیودها به زمین وصل می‌شوند چون آند آنها نسبت به کاتد مثبت‌تر نمی‌شود خاموش بوده و مشابه یک کلید قطع عمل می‌کنند و در خروجی ولتاژی نداریم. اما در زمان‌هایی که دیودها به ولتاژ ۵ ولت وصل می‌شوند چون در بایاس موافق قرار می‌گیرند مشابه کلید وصل عمل کرده و در خروجی تقریباً ولتاژی برابر با ولتاژ ورودی داریم.

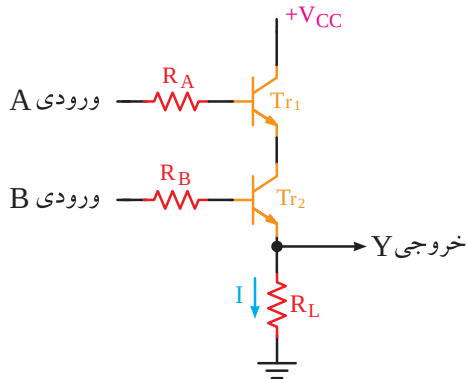


شکل ۲۳

از مقایسه مقادیر به دست آمده و در نظر گرفتن منطق دیجیتالی ($V \leftarrow 0$ و $V \leftarrow +5$) می‌توان نتیجه گرفت که عملکرد مدار، مشابه گیت OR است. در شکل ۲۳ مدار RDL گیت AND دو ورودی نشان داده شده است.

عملکرد مدار معادل RDL گیت AND را بررسی کرده و توضیح دهید.

پژوهش



شکل ۲۴- مدار معادل TTL گیت AND

در شکل ۲۴ مدار RTL معادل گیت AND با دو ورودی که با استفاده از دو ترانزیستور ساخته شده را مشاهده می‌کنید.

در این مدار دو ترانزیستور به صورت سری در مسیر $+V_{CC}$ و زمین قرار گرفته است. به علت قرار داشتن ولتاژ ترانزیستورها در وضعیت سری فقط در حالتی که ولتاژ H به هر دو ورودی A و B اعمال کنیم جریان برقرار خواهد شد و در نتیجه مقدار خروجی مدار در حالت H یا یک منطقی است. در غیر این صورت ولتاژ خروجی مدار در حالت L یا صفر منطقی خواهد بود.

جدول ۱۶- جدول صحت مدار معادل TTL گیت AND

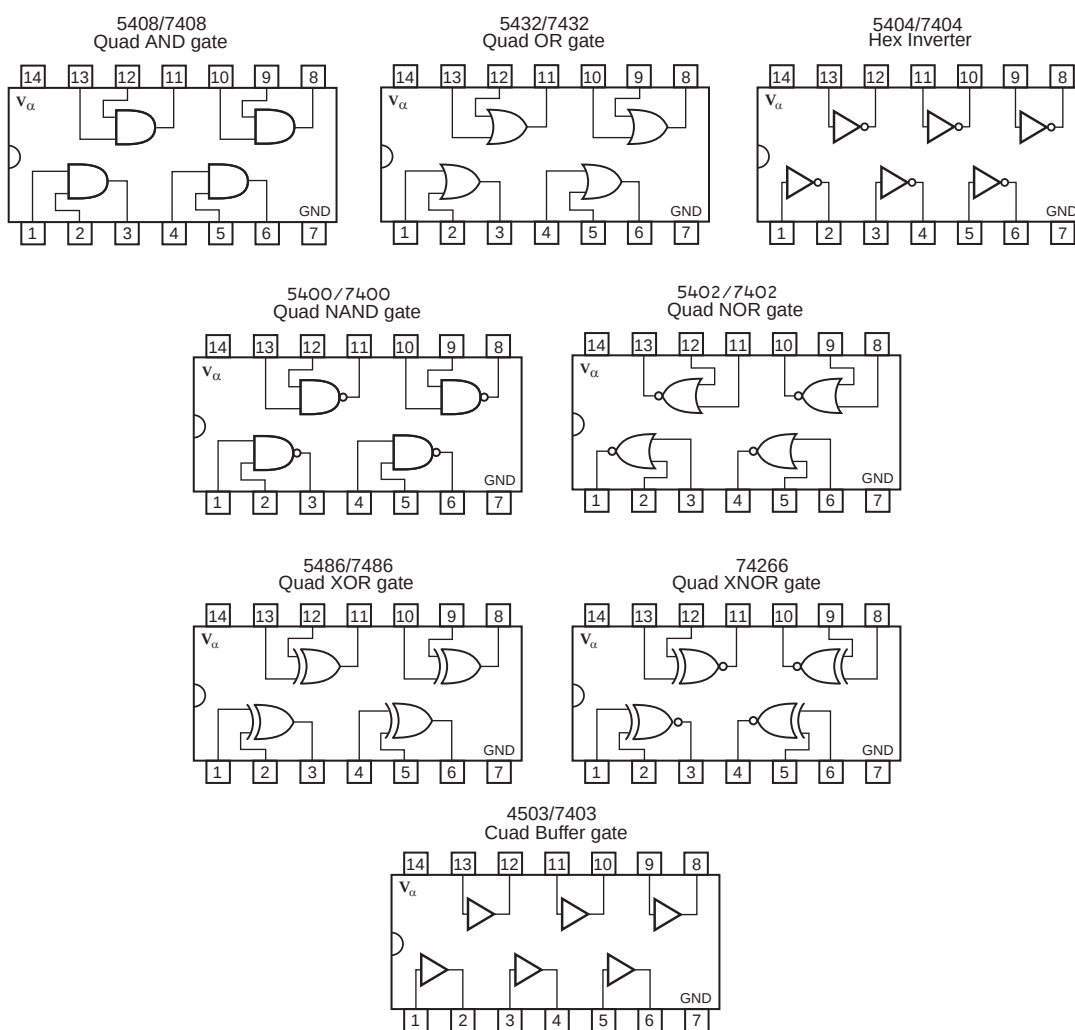
ورودی A	ورودی B	Tr_1	Tr_2	خروجی Y
L	L	قطع	قطع	L
L	H	قطع	وصل	L
H	L	وصل	قطع	L
H	H	وصل	وصل	H



مدار معادل TTL گیت OR را رسم کرده و در مورد عملکرد آن تحقیق کنید.

معرفی مشخصات پایه‌های^۱ (IC) گیت‌های منطقی

شکل ۲۵ نقشه داخلی تعدادی از IC های پرکاربرد در مدارهای دیجیتالی را نشان می‌دهد. در این شکل شماره پایه‌های IC، تعداد گیت‌ها و ارتباط گیت‌ها با پایه را مشاهده می‌کنید.



شکل ۲۵- نقشه داخلی IC های پرکاربرد

مدارهای منطقی

هدف از آشنایی با دروازه‌های منطقی استفاده از آنها در ساخت مدارهای منطقی است برای دستیابی به این مهم می‌بایست با نحوه طراحی و پیاده‌سازی خواسته‌های یک مسئله و چگونگی به‌کارگیری این دروازه‌ها آشنا شویم.

برای پیاده‌سازی هر مدار منطقی لازم است عملکرد آن در ازای حالت‌های مختلف ورودی‌ها بررسی شود. تشکیل جدول منظمی همچون «جدول صحت» در مباحث مدارهای منطقی یک ضرورت است. لذا در ابتدا با یکسری خصوصیات و نحوه تشکیل جدول صحت آشنا می‌شویم و سپس با ذکر مثال به بررسی عملکرد و پیاده‌سازی یک مدار منطقی می‌پردازیم.

■ جدول صحت:

برای تشکیل جدول صحت لازم است تا به نکات زیر توجه شود.

۱ به تعداد مجموع متغیرهای ورودی و خروجی باید ستون ایجاد شود.

۲ به تعداد حالات ممکنه مدار که از رابطه (2^n) به‌دست می‌آید باید سطر ایجاد شود. (n تعداد متغیرهای ورودی)

نکته

پس جدول صحت تابع ۱ متغیره دارای ۲ سطر، تابع ۲ متغیره دارای ۴ سطر، تابع ۳ متغیره دارای ۸ سطر و تابع ۴ متغیره دارای ۱۶ سطر و ... است (شکل ۲۶).



جدول صحت تابع ۱ متغیره

ورودی‌ها	خروجی
A	F
0	
1	

جدول صحت تابع ۳ متغیره

ورودی‌ها			خروجی
A	B	C	F
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

جدول صحت تابع ۴ متغیره

ورودی‌ها				خروجی
A	B	C	D	F
0	0	0	0	
0	0	0	1	
0	0	1	0	
0	0	1	1	
0	1	0	0	
0	1	0	1	
0	1	1	0	
0	1	1	1	
1	0	0	0	
1	0	0	1	
1	0	1	0	
1	0	1	1	
1	1	0	0	
1	1	0	1	
1	1	1	0	
1	1	1	1	

جدول صحت تابع ۲ متغیره

ورودی‌ها		خروجی
A	B	F
0	0	
0	1	
1	0	
1	1	

شکل ۲۶

شکل بیان عبارات منطقی

عبارات منطقی را در قالب یکی از دو شکل ۱- عبارت مینترم یا ۲- عبارت ماکسترم بیان می کنند. در اینجا به اختصار با هریک از آنها آشنا می شوید.

۱ عبارت مینترم (Minterm) تابع منطقی یا (عبارت حاصل جمع حاصل ضرب ها SOP - Sum of Product):

دو متغیر A و B را در نظر بگیرید طبق شکل ۲۷ این دو متغیر می توانند چهار حالت را در کنار یکدیگر داشته باشند. اگر این چهار حالت را با استفاده از عملگر AND (.) به صورت، $\bar{A}.\bar{B}$ ، $\bar{A}.B$ ، $A.\bar{B}$ ، $A.B$ نوشته و در نهایت جملات به دست آمده را به صورت حاصل جمع در کنار یکدیگر بنویسیم عبارت منطقی به دست خواهد آمد که اصطلاحاً به آن عبارت منطقی «عبارت مینترم» و به هر جمله آن نیز «جمله مینترم» گفته می شود.

جدول ۱۷

A	B	تابع منطقی هر سطر
0	0	$\bar{A}\bar{B}$
0	1	$\bar{A}B$
1	0	$A\bar{B}$
1	1	AB

عبارت مینترم را به شکل های دیگر نیز نشان می دهند. یکی از آنها براساس ارزش هر جمله و دیگری به جهت خلاصه نویسی، با استفاده از علامت (زیگما Σ) است که مفهوم حاصل جمع را دارد (شکل ۲۷).

$$F = \bar{A}\bar{B} + \bar{A}B + A\bar{B} + AB$$

$$F = m_0 + m_1 + m_2 + m_3$$

$$F = \sum (0, 1, 2, 3)$$

شکل ۲۷

جدول صحت و عبارت مینترم یک تابع سه متغیره را بنویسید.

فعالیت کلاسی



مثال



جدول ۱۸

A	B	F
0	0	1
0	1	0
1	0	1
1	1	1

با توجه به جدول صحت نشان داده شده در شکل ۲۸ عبارت مینترم آن را به شکل های مختلف بنویسید.

$$F = \bar{A}\bar{B} + \bar{A}B + AB$$

$$F = m_0 + m_2 + m_3$$

$$F = \sum (0, 2, 3)$$

شکل ۲۸

مثال



جدول ۱۹

ورودی‌ها			خروجی
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

 $\bar{A}\bar{B}C$ $A\bar{B}\bar{C}$ $A\bar{B}C$ ABC

به جدول ۱۹ توجه کنید. این جدول مربوط به خروجی یک سامانه فرضی است که با سه متغیر ورودی می‌تواند ۸ مینترم داشته باشد ($2^3 = 8$). همان‌طور که ملاحظه می‌شود، در خروجی چهارم مینترم تابع برابر با یک و چهار مینترم دیگر خروجی برابر صفر است. با استفاده از خروجی‌های ۱، تابع F را در ردیف‌هایی که برابر ۱ است به صورت مجموع مینترم‌ها می‌نویسیم.

$$F = \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C + ABC$$

درباره رابطه $F = \sum(1, 4, 5, 7)$ که مربوط به مینترم‌های تابع F مثال است، تحقیق کنید.

پژوهش



فعالیت کلاسی



جدول ۲۰

ورودی‌ها			خروجی
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

با توجه به جدول صحت ۲۰ تابع خروجی را به صورت مینترم بنویسید.

پایه‌سازی مدارهای منطقی

با توجه به توضیحات مقدماتی که داده شد حال به بررسی یک مثال می‌پردازیم به طوری که بیشتر هدف آشنایی با نحوه پایه‌سازی و ترسیم مدارهای منطقی است.



مداری طراحی کنید که بتوان با ۳ کلید دو وضعیتی A، B و C یک قفل را کنترل کرد. رمز قفل در حالتی باز می‌شود که فقط یک کلید بسته باشد (بسته بودن به مفهوم «یک» منطقی در نظر گرفته شده است).

حل: برای رسم مدار منطقی طرح مورد نظر لازم است ابتدا باید جدول صحت آن را تشکیل دهیم و سپس براساس آن عبارت منطقی را نوشته و در نهایت مدار را رسم کنیم.

جدول ۲۱- جدول صحت مربوط به مثال

شماره سطر	A	B	C	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	0
6	1	1	0	0
7	1	1	1	0

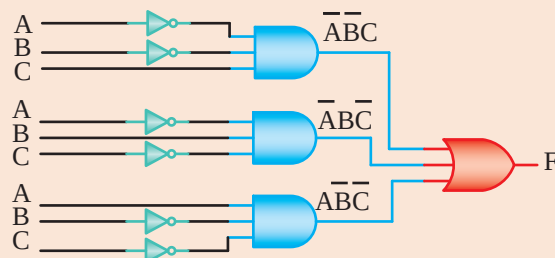
الف) ترسیم جدول صحت: ابتدا با توجه به تعداد متغیرها (۳ متغیر) و تعداد حالت‌ها ($2^3 = 8$) جدول صحت را مطابق جدول ۲۱ رسم می‌کنیم. در این جدول ستون ۱ شماره سطر، ستون‌های ۲ و ۳ و ۴ ورودی‌ها و ستون ۵ خروجی مدار یا همان F را نشان می‌دهد. براساس تعریفی که برای کار مدار شده است ستون خروجی (F) را تکمیل می‌کنیم. بنابر صورت مسئله، اگر فقط یکی از کلیدها (ورودی‌ها)، یک و بقیه صفر باشند، رمز قفل باز شده و در نتیجه ستون خروجی (F) باید یک شود.

ب) نوشتن تابع منطقی مدار: حال باید تابع منطقی را برای حالت‌هایی که مقدار خروجی یک است، بنویسیم. برای نوشتن تابع منطقی باید از جملات مینترم (حاصل جمع حاصل ضرب‌ها) استفاده کنیم.

$$F = \overline{A}\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C}$$

سطر ۴ سطر ۲ سطر ۱

ج) ترسیم مدار منطقی: با توجه به عبارت منطقی به دست آمده برای تابع F مدار شکل ۲۹ را طراحی و رسم می‌کنیم.

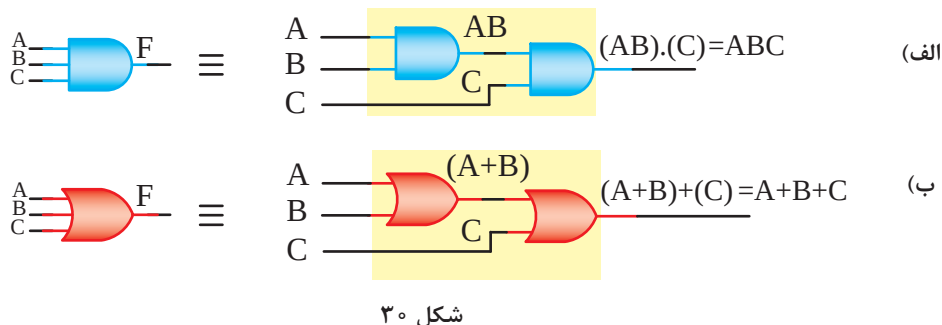


شکل ۲۹

همان طوری که مشاهده می کنید برای اتصال این مدار به ۱ گیت سه ورودی OR، ۳ گیت سه ورودی AND و ۶ گیت NOT و در مجموع به ۱۰ گیت نیاز است.

در صورتی که دروازه منطقی بیش از دو ورودی بخواهیم می توانیم از تعداد بیشتری گیت برای افزایش تعداد ورودی ها استفاده کنیم که به این کار «توسعه ورودی های گیت» گفته می شود.

به عنوان مثال تصاویر (الف و ب) شکل ۳۰ مدار معادل یک گیت AND و OR سه ورودی که با استفاده از دو گیت AND و OR دو ورودی ساخته شده را نشان می دهد.



در صورتی که بخواهیم مدار مثال قبل را با گیت های دو ورودی اتصال دهیم به چند گیت و چند IC نیاز است؟

بحث کنید



با ترسیم مدار و نوشتن تابع منطقی مشخص نمایید که آیا توسعه ورودی را برای گیت های منطقی (NAND ، NOR) نیز به همان صورت می توان انجام داد؟

پژوهش



کار عملی ۲



هدف: کسب مهارت در استخراج مشخصات فنی تراشه های (IC) دیجیتالی و بستن مدارهای منطقی

مواد، ابزار و تجهیزات: لوازم التحریر - اینترنت - رایانه - نرم افزار مولتی سیم یا هر نرم افزار مناسب دیگر - دیود نورانی LED قرمز و سبز - مقاومت های 150Ω ، $\frac{1}{4}$ وات، $10K\Omega$ ، $\frac{1}{4}$ وات و 330Ω ، $\frac{1}{4}$ وات - برد برد - میز آزمایشگاهی الکترونیک با تجهیزات استاندارد - کلید شستی کوچک (SPST Push Button) - کلید یک پل کوچک - برگه اطلاعات آی سی های دیجیتالی - آی سی های ۷۴۰۰، ۷۴۰۲، ۷۴۰۴، ۷۴۰۸، ۷۴۳۲، ۷۴۸۶

مراحل اجرای کار:

تمام آی سی هایی که برای آزمایش مدارهای دیجیتالی از آنها استفاده خواهید کرد از نوع TTL معمولی انتخاب شده اند. این آی سی ها به «سری ۷۴» مشهورند، زیرا دو رقم سمت چپ شماره سریال همه آنها با ۷۴ یا ۵۴ شروع می شود. ممکن است بعد از این دو رقم یک تا سه حرف آمده باشد و پس از این حروف نیز دو یا سه رقم دیگر و سرانجام یک یا دو حرف بعد از آن درج شده باشد که هر یک از آنها مفهوم خاصی دارد. در شکل ۳۱ این مفاهیم بیان شده است.



آی‌سی‌های دیجیتال در سری TTL و CMOS ساخته می‌شوند. در مورد شماره سریال، ولتاژ تغذیه و سایر موارد این نوع آی‌سی‌ها تحقیق کنید و نتایج را در قالب یک گزارش ارائه دهید.

SN ۷۴ LS ۰۴ N				
نوع بسته‌بندی: DIP سرامیکی - JDIP پلاستیکی - N سرامیکی مسطح - W	معرف کار مدار، به عنوان مثال در این آی‌سی ۶ دروازه NOT وجود دارد.	مشخصات آی‌سی‌های خانواده TTL (برای اطلاع دقیق به مثال‌های زیر توجه کنید).	نوع تجاری برای کار در دمای صفر تا ۷۰ درجه سانتی‌گراد	پیشوند انتخابی کارخانه سازنده

مثال‌هایی مربوط به مشخصات آی‌سی‌های خانواده TTL

شماره آی‌سی	نماد	مفهوم
۷۴۰۴	بدون حرف	استاندارد
۷۴H۰۴	H = High Speed	سریع
۷۴L۰۴	L = Low Power	کم‌مصرف
۷۴S۰۴	S = Schottky	نوع شاتکی
۷۴SL۰۴	LS = Low Power Schottky	کم‌مصرف شاتکی
۷۴AS۰۴	AS = Advsnce Schottky	شاتکی اصلاح شده
۷۴ALS۰۴	ALS = Advance Low Power Schottky	شاتکی کم‌مصرف اصلاح شده

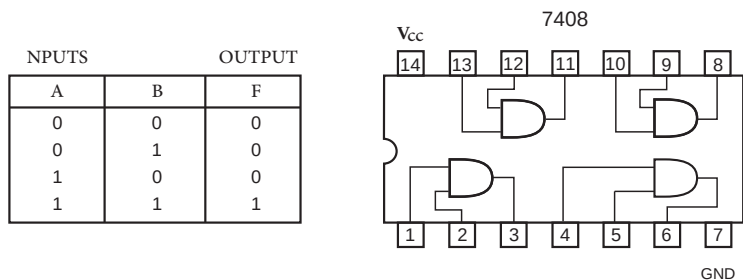
شکل ۳۱- مشخصات آی‌سی‌های TTL

جدول ۲۲

ردیف	شماره آی‌سی	مشخصات فنی از روی شماره
۱		
۲		
۳		

۱ سه نمونه آی‌سی سری ۷۴ یا ۵۴ را دریافت کنید و با مراجعه به شکل ۳۱ شماره برخی از مشخصات آنها را استخراج و در جدول ۲۲ یادداشت کنید.

۲ با توجه به شکل ۳۲ مشخصات آی سی ۷۴۰۸ را از نظر تعداد گیت و جدول صحت هر گیت توضیح دهید.



INPUTS		OUTPUT
A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

۳ با مراجعه به برگه اطلاعات آی سی ۷۴۰۸ در شکل ۳۲ مقادیر مجاز ماکزیمم و مینیمم ولتاژ و جریان و محدوده درجه حرارت کار آی سی ۷۴۰۸ را در جدول ۲۳ یادداشت کنید.

Recommended Operating Conditions

	SN 74LS08			
	MIN	NOM	MAX	Unit
Vcc Supply Voltage	4.75	5	5.25	V
High - Level input voltage	2			V
Low - Level input voltage			0.8	V
High - Level output current			-0.4	mA
Low - Level output current			8	mA
Operating free-air temperature	0		70	C

شکل ۳۲- برگه اطلاعات آی سی ۷۴۰۸

جدول ۲۳

Symbol نماد	Parametr مشخصه		مقادیر Min Non Max		
			Min	Non	Max
VCC	Supply Voltage	ولتاژ تغذیه			
VIH	High Level Input Voltage	ولتاژ حالت H ورودی			
VIL	Low Level Input Voltage	ولتاژ حالت L ورودی			
IOH	High Level Output Current	جریان خروجی در حالت سطح بالا			
IOL	Low Level Output Current	جریان خروجی در حالت سطح پایین			
TA	Free Air Operating Temperature	درجه حرارت کار در هوای آزاد			

۴ مدار داخلی آی سی های گیت منطقی را از نظر تعداد گیت و جدول صحت بررسی کنید. جدول ۲۳ را کامل کنید

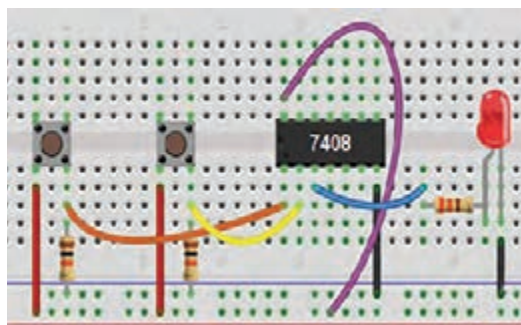
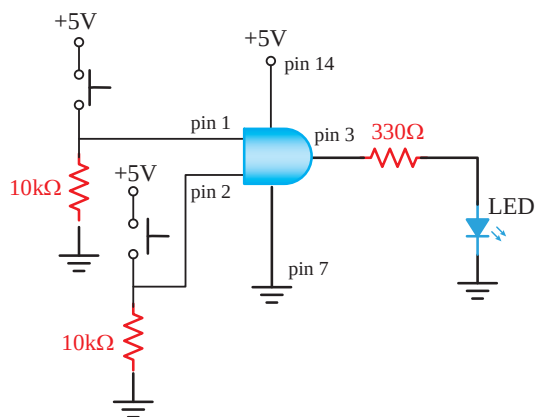
جدول ۲۴- شماره آی سی ها و جدول صحت

IC۷۴۳۲			IC۷۴۰۰			ردیف	شماره آی سی	نوع گیت و تعداد آن	تابع خروجی
A	B	F	A	B	F				
0	0					۱	۷۴۳۲		
0	1					۲	۷۴۰۰		
1	0					۳	۷۴۰۲		
1	1					۴	۷۴۸۶		
IC۷۴۰۲			IC۷۴۸۶			۵	۷۴۰۴		
A	B	F	A	B	F				

IC۷۴۰۴		
A	B	F

شماره آی سی ها جهت آشنایی با مدار داخلی آنهاست و نیاز به حفظ کردن آنها و شماره پایه ها نیست. در صورت استفاده باید به برگه اطلاعات هر یک در Data Book مراجعه شود.

۵ با استفاده از آی سی ۷۴۰۸ مدار شکل ۳۳ را روی بردبرد ببندید.



شکل ۳۳- مدار گیت AND

کلیدهای A و B را به صورت جداگانه یا با هم تغییر وضعیت دهید و به وضعیت روشن شدن LED توجه کنید. نتایج حاصل را در جدول ۲۴ بنویسید.

۶ با استفاده از جدول ۲۴ جدول صحت گیت AND را رسم کنید و در مورد عملکرد مدار توضیح دهید.

.....

.....

.....

جدول ۲۵

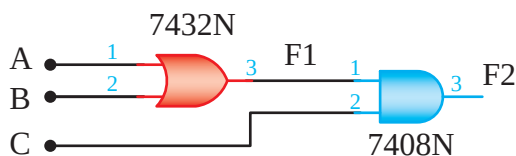
وضعیت ورودی‌ها		حالت خروجی
A	B	F = LED
قطع	قطع	
قطع	وصل	
وصل	قطع	
وصل	وصل	

مدار سایر گیت‌های منطقی را ببندید و جدول صحت آنها را به دست آورید.

۷ جدول صحت مدار منطقی شکل ۳۴ در جدول ۲۶ بنویسید.

جدول ۲۶- جدول صحت مدار منطقی

وضعیت ورودی‌ها			حالت خروجی	
A	B	C	F _۱	F _۲



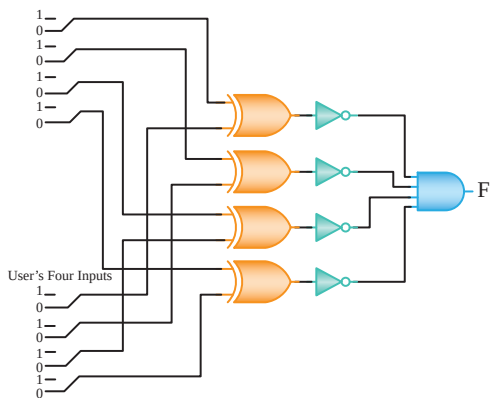
شکل ۳۴- مدار منطقی

F_۱ =

F_۲ =

۸ تابع منطقی F_۱ و F_۲ در شکل ۳۴ را بنویسید.

Initial Four Inputs



شکل ۳۵- مدار قفل رمز چهاربیتی

۹ مدار شکل ۳۵ مدار منطقی در ورودی یک آزمایشگاه تحقیقاتی با قفل رمز چهاربیتی که دارای قابلیت تنظیم رمز ورود است را نشان می‌دهد. نحوه عملکرد آن بدین صورت است که هرگاه فردی که در پشت در قرار دارد کد چهاربیتی مشابه کد تنظیمی قفل را وارد کند خروجی مدار یک شده و در باز می‌شود و در غیر این صورت امکان باز شدن در وجود ندارد.

توضیح: برای افزایش امنیت، مسئول آزمایشگاه می‌تواند از بین ۱۶ رمز متفاوت که می‌توان برای در تعریف کرد هر زمان که نیاز باشد یکی را انتخاب کرده و برای در تعریف کند.

۱۰ مدار را روی برد برد اتصال داده و راه‌اندازی کنید.

۱۱ با تنظیم رمز در، برای یک کد چهاررقمی خاص و تشکیل جدول صحت طرز کار مدار را بررسی کنید.

ارزشیابی شایستگی اجرای مدار عملگرهای منطقی

نام و نام خانوادگی		شماره ملی	تاریخ ارزشیابی:	
کد حرفه	۸۲۱۲۱۲	حرفه :	مونتاز کاران تجهیزات و وسایل برقی و الکترونیکی	کد پیمان/پودمان ۸۲۱۲۱۲۰۳۲
کد وظیفه شغل	۸۲۱۲۱۲۰۳	وظیفه شغل:	ساخت مدارهای دیجیتال و میکروکنترلر	سطح صلاحیت ۲
کد کار	۸۲۱۲۱۲۰۳۲۱	کار	اجرای مدار عملگرهای منطقی	سطح شایستگی مهارت

ردیف	مراحل کار	شرایط انجام کار (ابزار، مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص ها/داوری/نمره دهی)	نتایج ممکن	نمره کسب شده
۱	تبدیل صحیح مبناهای عددی	دست نویس یا نرم افزار آموزشی تبدیل مبناهای (دهدهی، دودویی، هشت تایی، هگزادسیمال)، دفترچه یادداشت، قلم، تایمر و زمان ۱۵-۲۰ دقیقه	بررسی تبدیل اعداد، استفاده از چکلیست، آزمون کتبی یا شفاهی	تبدیل صحیح و سریع اعداد	۳	
				تبدیل ها صحیح با اشتباهات جزئی	۲	
				تبدیل با اشتباهات زیاد یا ناقص	۱	
۲	شناسایی و کار با آی سی های منطقی، پیاده سازی توابع منطقی با گیت ها	برد برد یا برد آموزشی، آی سی گیت منطقی، منبع تغذیه، سیم های رابط، LED و مقاومت، برکه تمرین تابع منطقی و زمان ۳۰-۲۰ دقیقه	مشاهده عملکرد مدار و جدول درستی، چکلیست ارزیابی مدار و اتصالات، پرسش شفاهی	پیاده سازی کامل و صحیح تابع منطقی با عملکرد درست	۳	
				پیاده سازی با خطای جزئی ولی خروجی قابل قبول	۲	
				پیاده سازی نادرست و عدم تطابق خروجی با تابع منطقی	۱	
۳	اجرای عملی مدار منطقی و عیب یابی و تحلیل مدار	برد برد یا برد آموزشی، آی سی گیت منطقی، منبع تغذیه، سیم های رابط، LED و مقاومت، مولتی متر، برکه مدار و زمان ۳۰-۲۵ دقیقه	مشاهده عملکرد مدار، بررسی خروجی ها، بررسی دقیق اتصالات، چکلیست ارزیابی و پرسش شفاهی	اجرا و عیب یابی صحیح مدار و تحلیل دقیق عملکرد	۳	
				اجرای مدار با رفع خطاهای جزئی و تحلیل قابل قبول	۲	
				عدم توانایی در راه اندازی یا عیب یابی و تحلیل نادرست مدار	۱	
	شایستگی های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:	آزمایشگاه مدارهای منطقی، تجهیزات آموزشی و راهنمای کار، تجهیزات کارگاهی و حفاظت فردی و دستورالعمل های ایمنی و بهداشت حرفه ای	مشاهده رفتار هنرجو در حین انجام فعالیت، میزان رعایت اصول ایمنی و نظم، فعالیت های گروهی، مدیریت زمان و مسئولیت پذیری	رعایت کامل ایمنی، بهداشت و محیط زیست، رفتار حرفه ای، نظم و مسئولیت پذیری بالا	۲	
				رعایت نسبی ایمنی و بهداشت با خطاهای جزئی، رفتار کاری مناسب	۱	

☐ بلی

☐ خیر

ارزشیابی کار (شایستگی انجام کار)

توضیحات:

۵- معیار شایستگی انجام کار:

- کسب حداقل نمره ۲ در مراحل بحرانی شامل تبدیل صحیح مبناهای عددی، پیاده سازی درست توابع منطقی با گیت ها و اجرای عملی مدار منطقی همراه با عیب یابی و تحلیل مدار است.
- کسب حداقل نمره ۲ از بخش شایستگی های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش
- کسب حداقل میانگین ۲ از مراحل کار

۶- تعریف سطوح شایستگی: صرف نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه ای انجام می شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

- مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه ریزی و تحلیل، پاسخ گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت ها
- تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب یابی، هدایت و راهنمایی دیگران.

واحد یادگیری ۲: اجرای مدارهای منطقی

آیا تا به حال فکر کرده‌اید

- ۱ چرا قبل از ترسیم و پیاده‌سازی یک مدار منطقی، باید تابع منطقی را ساده کنیم؟
- ۲ ساده‌سازی عبارات منطقی چه تأثیری بر حجم و هزینه مدار دارد؟
- ۳ از چه روش‌هایی می‌توان برای ساده‌سازی توابع منطقی استفاده کرد؟
- ۴ نرم‌افزار Multisim چگونه می‌تواند در طراحی و اجرای مدارهای دیجیتال به ما کمک کند؟

قبل از ترسیم و پیاده‌سازی سخت‌افزاری یک مدار منطقی، لازم است تابع منطقی مورد نظر را ساده کنیم تا به عبارتی کوتاه‌تر و مناسب‌تر دست یابیم. یک عبارت یا تابع منطقی ساده را می‌توان با استفاده از توابع منطقی اصلی یا ترکیبی از آنها پیاده‌سازی کرد.

دلایل ساده‌سازی عبارات منطقی به اختصار عبارت‌اند از:

- ۱ کاهش حجم مدار
- ۲ کاهش تلفات مدار
- ۳ کاهش هزینه ساخت مدار

برای ساده‌سازی عبارات منطقی می‌توان از روش‌های مختلفی مانند قوانین جبر بول و نقشه کارنو استفاده کرد. علاوه بر این، در این واحد یادگیری، هنرجویان با ساده‌سازی توابع منطقی به کمک نرم‌افزار Multisim که بر پایه دو روش یاد شده عمل می‌کند، آشنا می‌شوند و توانایی و مهارت طراحی و اجرای پروژه‌های ساده و کاربردی مدارهای دیجیتال را به دست می‌آورند. همچنین، شایستگی‌های غیرفنی مانند رعایت اصول ایمنی و نظم، کار گروهی، مدیریت زمان و مسئولیت‌پذیری نیز در آنها تقویت می‌شود.

استاندارد عملکرد

ساده‌سازی و پیاده‌سازی توابع منطقی با استفاده از جبر بول، نقشه کارنو و گیت‌های اصلی

1 قوانین جبر بول:

جبر بول مدلی ریاضی برای مدارهای منطقی است که چگونگی عملکرد یک مدار دیجیتالی را با زبان ریاضی تعریف شده خاص خود بیان می‌کند. با استفاده از قوانین جبر بول، مدل ریاضی ساده و قابل استفاده‌ای برای انواع مدارهای منطقی، مدارهای ترکیبی و ترتیبی دیجیتالی ارائه می‌شود. با استفاده از این روابط می‌توانیم به موارد زیر دست یابیم.

■ روابط بین خروجی‌ها و ورودی‌های مدار به صورت ساده‌ترین شکل ریاضی.

■ اجرای واقعی مدارهای کاربردی و عملی و آموزش آنها به ساده‌ترین شکل.

■ بیان صحیح عملکرد منطقی مدارها.

برای فراگیری جبر بول و روابط آن لازم است برخی از مفاهیم را به خوبی یاد بگیریم. بدیهی است بدون فراگیری این مفاهیم نمی‌توانیم مدل‌های ریاضی جبر بول را به کار ببریم یا آنها را ساده کنیم. بسیاری از این موارد با مفاهیم پایه‌ای ریاضی تطابق داشته و برخی از آنها فقط مربوط به جبر بول است. در ادامه به اختصار چند قانون پرکاربرد جبر بول را در جدول ۲۷ با ذکر مثال‌هایی مشاهده می‌کنید.

جدول ۲۷- قوانین پرکاربرد جبر بول

توضیح	قوانین جبر بول	تشریح
با عملگر OR	عضو خنثی (inactive)	$0 + A = A$ مثال: $0 + AB + \overline{CD} = AB + \overline{CD}$
با عملگر AND	عضو خنثی (inactive)	$1 \cdot A = A$ مثال: $1 \cdot (AB + CD) = AB + CD$
با عملگر OR با عملگر AND	قانون یکسانی یا عینیت (Identity)	$A + A + A = A$ $A \cdot A \cdot A = A$
با عملگر OR با عملگر AND	قانون متمم (Complement)	$A + \overline{A} = 1$ $A \cdot \overline{A} = 0$
با عملگر OR با عملگر AND	قانون جابه‌جایی (Relocation)	$A + B + C = A + C + B = C + A + B = B + C + A = \dots$ $A \cdot B \cdot C = B \cdot C \cdot A = C \cdot A \cdot B = B \cdot C \cdot A = \dots$
با عملگر OR با عملگر AND	قانون اتحاد (association)	$A + (B + C) = (A + B) + C$ $A \cdot (B \cdot C) = (A \cdot B) \cdot C$
توزیع AND در OR توزیع OR در AND	قانون توزیع (Distribution)	$A \cdot (B + C) = A \cdot B + A \cdot C$ $A + B \cdot C = (A + B) \cdot (A + C)$
عملگر NOT روی OR عملگر NOT روی AND	قانون دمورگان (Dmorgan)	$\overline{A + B} = \overline{A} \cdot \overline{B}$ $\overline{A \cdot B} = \overline{A} + \overline{B}$

فعالیت کلاسی



با کمک و راهنمایی مربی خود در مورد هریک از قوانین جبر بول با هم گروهی خود بحث کنید.

فعالیت کلاسی



مدار معادل کلیدی هریک از قوانین جبر بول را رسم کنید.

مثال ۱



شکل ساده شده عبارت مقابل با استفاده از قضایای جبر بول چیست؟

$$(A\bar{B} + A\bar{B}).1 = ?$$

$$(A\bar{B} + A\bar{B}).1 = (A\bar{B}).1 \\ = A\bar{B}$$

مثال ۲



تابع داده شده زیر را با استفاده از قضایای جبر بول ساده کنید.

$$A\bar{A} + BC = ?$$

$$A\bar{A} + BC = 0 + BC \\ = BC$$

مثال ۳



تابع داده شده زیر را با استفاده از قضایای جبر بول ساده کنید.

$$F = \overline{A + B + \bar{A}B}$$

$$= \bar{A}\bar{B} + \bar{A}B$$

$$= \bar{A}(\bar{B} + B)$$

$$= \bar{A}.1$$

$$= \bar{A}$$

مثال ۴



تابع داده شده زیر را با استفاده از قضایای جبر بول ساده کنید.

$$F = \overline{A + B + C + A\bar{B} + \bar{B}C}$$

$$= \bar{A}\bar{B}\bar{C} + \bar{B}(A + C)$$

$$= \bar{B}(\bar{A}\bar{C} + A + C)$$

$$= \bar{B}(\overline{A + C} + A + C)$$

$$= \bar{B}((\overline{A + C}) + (A + C))$$

$$= \bar{B}.1$$

$$= \bar{B}$$



در مورد تساوی طرفین عبارت مقابل تحقیق کنید.

$$(A + B)(A + C) = A + BC$$

$$(A + B)(A + C) = AA + AC + AB + BC$$

$$= A + AC + AB + BC$$

$$= A(1 + C) + AB + BC$$

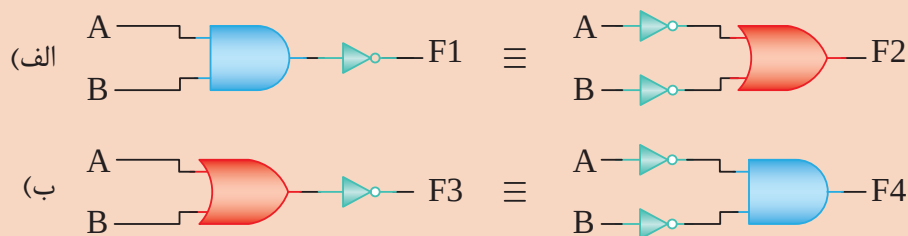
$$= A(1 + B) + BC$$

$$= A.1 + BC$$

$$= A + BC$$



با نوشتن عبارات هر سمت مدارهای منطقی، در مورد معادل بودن F_1 با F_2 و F_3 با F_4 تحقیق کنید. نتایج به دست آمده بیانگر کدام قانون است؟



شکل ۳۶



توابع منطقی داده شده را در نظر گرفته و به موارد زیر پاسخ دهید.

(الف) مدار منطقی تابع را قبل از ساده سازی رسم نمایید. (ب) توابع را با استفاده از قضایای جبر بول ساده کنید. (ج) مدار منطقی تابع را بعد از ساده سازی رسم نمایید. (د) با تشکیل جدول صحت در مورد عملکرد هر دو مدار تحقیق کنید.

(الف) $F_1 = A\bar{B} + \bar{A}B + AB$

(ب) $F_2 = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + AB\bar{C} + A\bar{B}C$

(ج) $F_3 = \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}C + \bar{A}BC + A\bar{B}C$

۲ جدول (نقشه) کارنو (Karnaugh map)

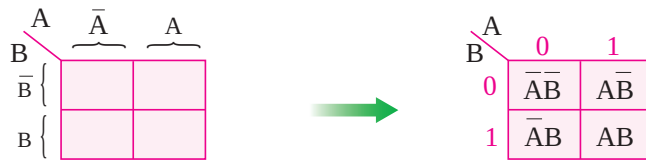
یکی دیگر از روش های ساده سازی عبارات منطقی استفاده از جدول (نقشه) کارنو است. نقشه کارنو در واقع شکل تغییر یافته جدول صحت است که در آن ارتباط بین متغیرها کنار هم نشان داده می شوند. برای ترسیم جدول کارنو به نکات زیر باید توجه کرد.

- ۱- تعداد خانه های جدول از رابطه 2^n به دست می آید. (n - تعداد متغیرها (ورودی ها))
- ۲- هر متغیر (مثلاً A) باید نصف خانه های جدول و نصف دیگر را نفی آن ($\bar{A}$) تشکیل دهد.
- ۳- هر دو متغیر باید با یکدیگر در ($\frac{1}{4}$) خانه های جدول اشتراک داشته باشند.

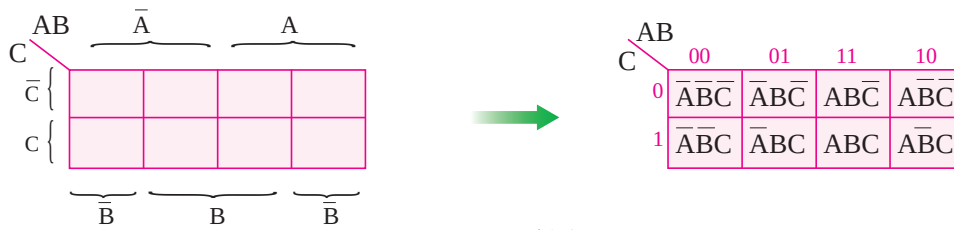
پس باتوجه به موارد اشاره شده، جدول صحت عبارات منطقی یک، دو، سه و چهار متغیره را به صورت های زیر می توان رسم کرد (شکل ۳۷).



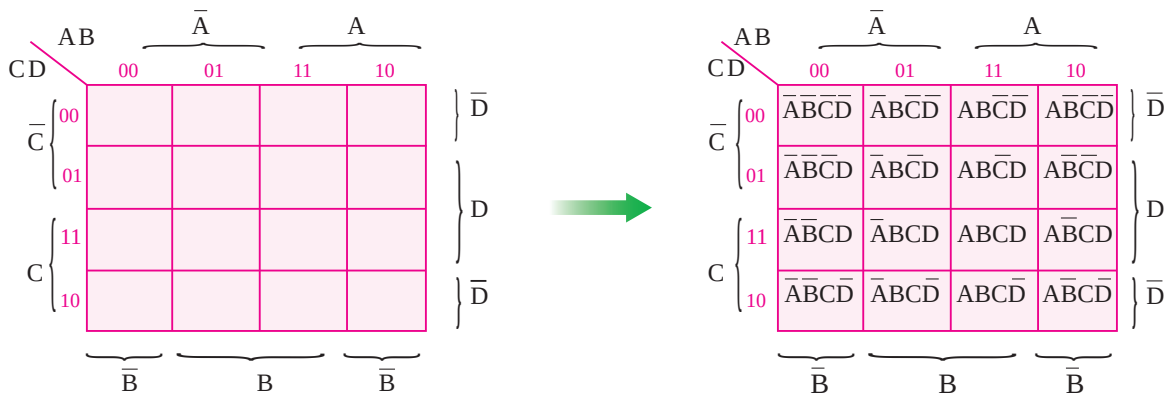
جدول کارنو یک متغیره



جدول کارنو دو متغیره



جدول کارنو سه متغیره

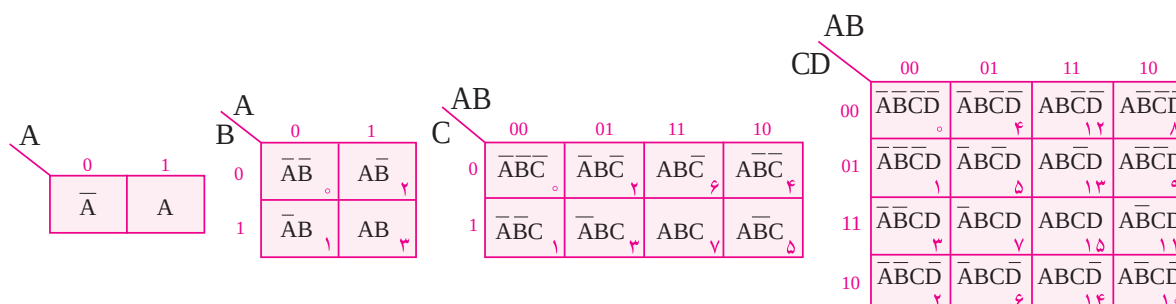


جدول کارنو چهار متغیره

شکل ۳۷

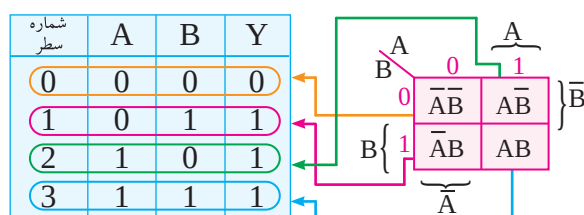
با کمی دقت و ارزش گذاری برای هریک از متغیرهای داخل جدول کارنو ($\bar{A} = 0$ و $A = 1$) می توان به یک سری اعداد باینری دست یافت که با تبدیل آنها به اعداد دهدهی می توان نتیجه گرفت که هریک از خانه های جدول

کارنو دارای ارزش هستند که از این پس آن خانه‌ها را با شماره‌های آن می‌شناسیم (شکل ۳۸).



شکل ۳۸- ارزش (شماره) خانه‌های جداول کارنو ۱ تا ۴ متغیره

شماره هر خانه جدول کارنو متناظر با شماره سطر جدول صحت تابع است (شکل ۳۹).



شکل ۳۹

اگر به نقشه کارنو خوب دقت کنید، می‌بینید که از هر خانه به خانه مجاور در جهت افقی یا عمودی فقط یک متغیر تغییر می‌کند.

■ ساده‌سازی عبارات منطقی با جدول کارنو

برای ساده‌سازی به کمک جدول کارنو کافی است با توجه به عبارت منطقی داده شده ارزش عددی هر جمله آن را مشخص کرده و به ازای هر جمله عبارت منطقی عدد یک را در خانه مورد نظر قرار دهیم. برای ساده‌سازی باید مجموعه یک‌های مجاور هم را که به صورت عمودی یا افقی کنار هم قرار گرفته‌اند را به صورت یک گروه در نظر بگیریم. در نهایت نیز لازم است تا موقعیت گروه تشکیل شده را نسبت به متغیرهای جدول بررسی کنیم و ببینیم که گروه ایجاد شده به طور کامل کدام یک از متغیرها را دربر می‌گیرد. لازم به ذکر است که تعداد یک‌های هر گروه براساس رابطه (2^n) می‌تواند ۱، ۲، ۴، ۸ و... باشد. هرچه تعداد یک‌های موجود در هر گروه بیشتر باشد عبارت ساده شده مناسب‌تری به دست می‌آید که دیگر نیاز به ساده‌سازی نیست. در ادامه با ذکر چند مثال با جزئیات ساده‌سازی توابع منطقی بیشتر آشنا خواهید شد.

تابع $F = \overline{A}B + A\overline{B} + AB$ را با استفاده از نقشه کارنو ساده کنید.

$$F = \overline{A}B + A\overline{B} + AB$$

عبارت منطقی ←

ارزش باینری ←

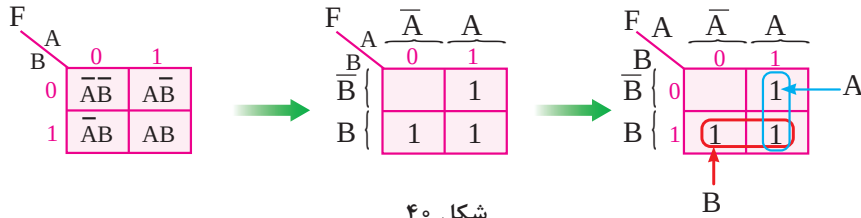
شماره خانه ←

۰۱ ۱۰ ۱۱

۱ ۲ ۳

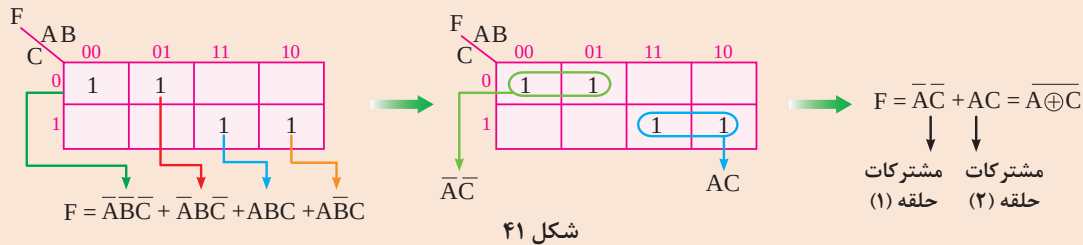
مثال ۶





با توجه به گروه‌های تشکیل شده می‌توان نتیجه گرفت عبارت ساده شده تابع F به صورت $F = A + B$ خواهد شد.

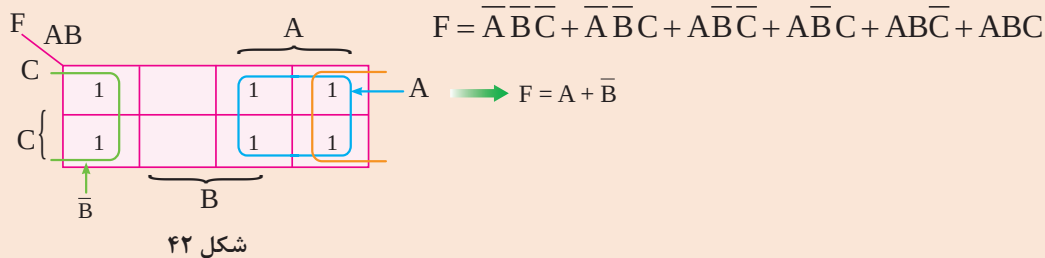
تابع $F = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + A\bar{B}\bar{C} + A\bar{B}C$ را به کمک جدول کارنو ساده کنید.



مثال ۷



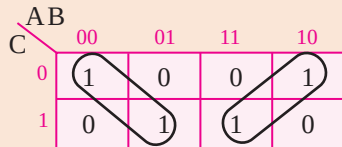
ساده‌ترین شکل تابع F را با استفاده از جدول کارنو به دست آورید.



مثال ۸



در جدول کارنو هیچ‌گاه یک‌های مجاور هم را به صورت مورب نمی‌توان به صورت گروه در نظر گرفت (شکل ۴۳).

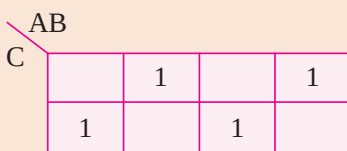


جدول کارنو با خانه‌های مشترک مورب
شکل ۴۳

تذکر



جدول (نقشه) کارنو داده شده شکل ۴۴ را ساده کرده و عبارت منطقی آن را بنویسید. چون یک‌ها مجاور هم نیستند امکان تشکیل گروه نیست و عبارت ساده شده‌ای به دست نمی‌آید.



شکل ۴۴

$$F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$$

مثال ۹





۱ هریک از توابع داده شده زیر را در نظر گرفته و شکل ساده شده هریک را به کمک روش‌های زیر به دست آورید.

الف) کمک قضایای جبر بول ب) به کمک قضایای جبر بول

الف) $F = (A, B, C) = \bar{A}\bar{B}C + A\bar{B}\bar{C} + \bar{A}B\bar{C} + ABC + A\bar{B}C$

ب) $F = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$

ج) $F = (A, B, C, D) = \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + AB\bar{C}D + \bar{A}\bar{B}C\bar{D} + \bar{A}B\bar{C}\bar{D} + A\bar{B}C\bar{D} + \bar{A}BC\bar{D}$

د) $F = (A, B, C, D) = \bar{A}\bar{B}\bar{C}D + A\bar{B}\bar{A}\bar{A} + \bar{A}B\bar{C}D + \bar{A}B\bar{C}\bar{D} + \bar{A}BC\bar{D}$

۲ مدار منطقی هریک از توابع داده شده در تمرین ۱ را در حالات خواسته شده رسم کنید.

الف) قبل از ساده‌سازی ب) پس از ساده‌سازی

۳ با تشکیل جدول صحت، درستی عملکرد هر دو مدار (ساده نشده و ساده شده) را بررسی کنید.



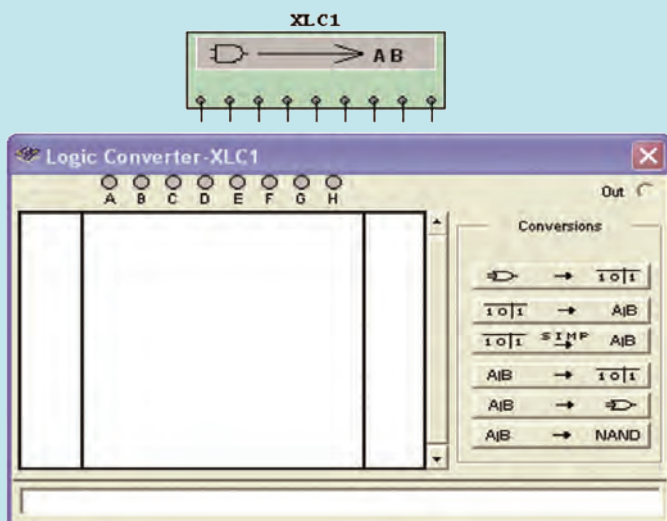
ساده‌سازی تابع منطقی با نرم‌افزار

هدف: کسب شایستگی ساده‌سازی توابع منطقی و پیاده‌سازی تابع ساده شده با گیت‌های منطقی

مواد، ابزار و تجهیزات: رایانه - نرم‌افزار مولتی‌سیم - نوشت افزار

مراحل اجرای کار:

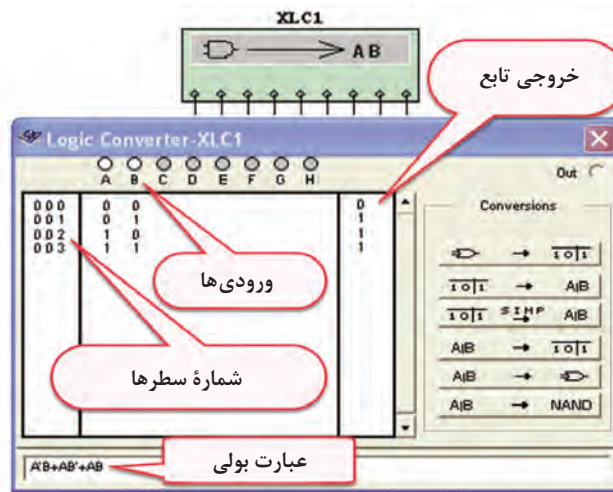
در نرم‌افزار مولتی‌سیم دستگاهی به نام Logic Converter یا مبدل منطقی وجود دارد که می‌توانید عملیات مختلفی را با آن انجام دهید. به عنوان مثال اگر جدول صحت گیت را بنویسید، با فعال کردن یکی از دکمه‌های این دستگاه عبارت بولی گیت مورد نظر نوشته می‌شود و همچنین می‌توانید مداری را طراحی کنید و جدول صحت و عبارت منطقی آن را مشاهده نمایید.



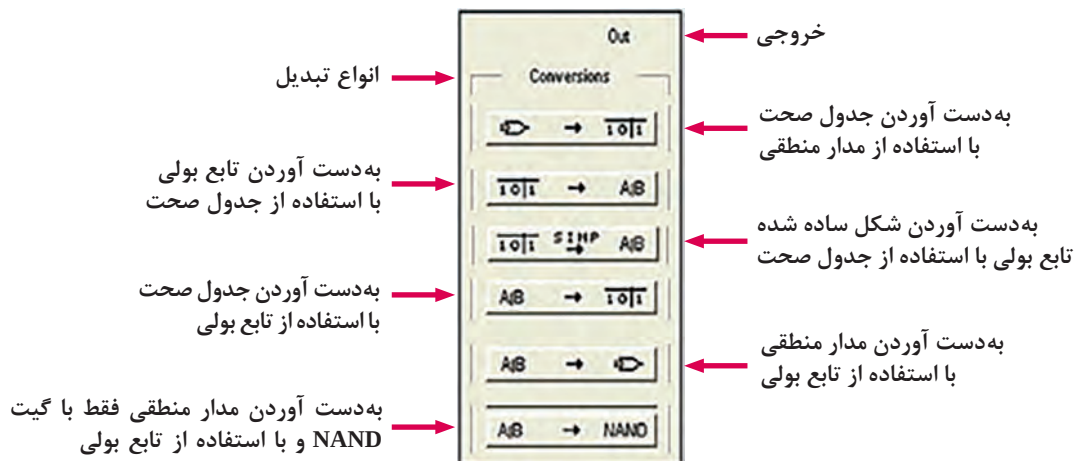
شکل ۴۵- موقعیت دستگاه مبدل منطقی در نرم‌افزار

1 دستگاه Logic Converter را بر روی صفحه آزمایشگاه مجازی بیاورید. ورودی‌های مدار را از قسمت‌های A, B و... انتخاب کنید.

اگر دو ورودی را انتخاب کنید، چهار سطر و دو ستون برای ورودی‌ها شکل می‌گیرد و به ترتیب اعداد صفر و یک منطقی را مطابق اعداد باینری به ورودی‌ها اختصاص می‌دهد. در این جدول، ستون اول از سمت چپ شماره سطر و ستون دوم در وسط مربوط به ورودی‌ها و ستون سوم از سمت چپ خروجی را نشان می‌دهد. با کلیک کردن بر روی علامت سؤال در ستون خروجی جدول، یک بار عدد صفر و با کلیک مجدد عدد یک به خروجی هر سطر اختصاص داده می‌شود. به این ترتیب می‌توانید خروجی دلخواه خود را بنویسید. به عنوان مثال برای گیت OR به ترتیب اعداد باینری ۰، ۱، ۱ و ۱ را برای خروجی تعیین کنید. با فعال کردن دکمه $\overline{A} \cdot \overline{B} \rightarrow A \cdot B$ عبارت بولی مربوط به جدول صحت بر روی کادر پایین صفحه نوشته می‌شود. در شکل ۴۶ کلیدهای مربوط به تبدیل را در دستگاه مبدل منطقی مشاهده می‌کنید.



شکل ۴۶- جدول صحت و عبارت بولی گیت منطقی OR



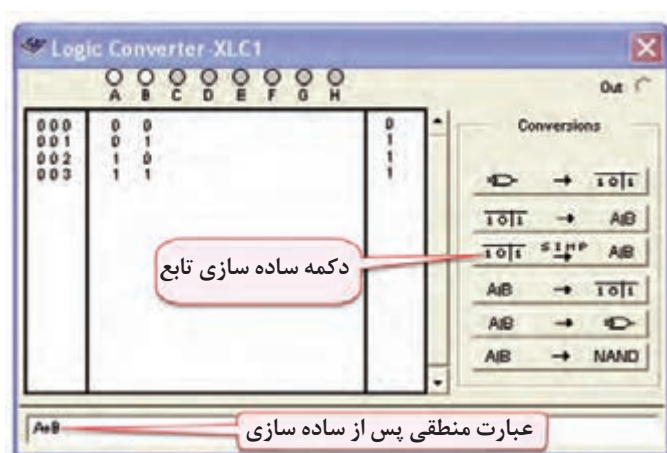
شکل ۴۷- کلیدهای مربوط به دستگاه مبدل منطقی

با توجه به شکل ۴۷ درمی‌یابیم که دستگاه مبدل منطقی می‌تواند کاربردهای گسترده‌ای داشته باشد که در ادامه به عملکرد این دستگاه خواهیم پرداخت.

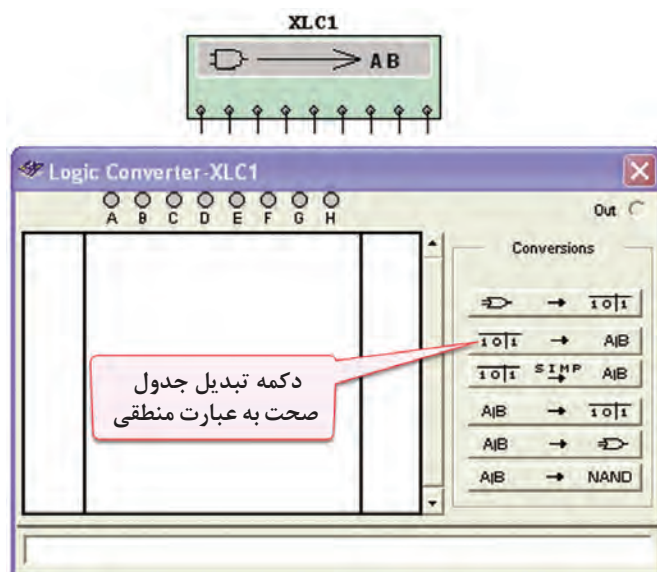
۲ ساده‌سازی تابع منطقی با دستگاه مبدل منطقی (Logic Converter)

یکی دیگر از قابلیت‌های دستگاه مبدل منطقی، به‌دست آوردن عبارت منطقی گیت یا مدار منطقی است که جدول آن را کامل کرده‌اید.

پس از اینکه جدول مدار گیت را تکمیل کردید، دکمه تبدیل جدول به عبارت را فعال کنید. عبارت منطقی مربوط به مدار در کادر پایین صفحه دستگاه ظاهر می‌شود. در شکل ۴۸ با فعال کردن دکمه *simplify* شکل ساده شده عبارت منطقی در کادر مربوطه ظاهر خواهد شد. در شکل ۴۹ موقعیت این دکمه و عبارت ساده شده گیت OR را مشاهده می‌کنید.



شکل ۴۸- کلید تبدیل ساده‌سازی جدول صحت و تابع



شکل ۴۹- کلید تبدیل جدول صحت به تابع

۳ هر یک از توابع منطقی زیر را با دستگاه مبدل منطقی (Logic Converter) ساده کرده و در جدول ۲۸ بنویسید.

جدول ۲۸

$F_1 =$	۱- تابع اصلی جدول صحت F_1 ۲- تابع ساده شده F_1
$F_1 =$	
$F_2 =$	۱- تابع اصلی جدول صحت F_2 ۲- تابع ساده شده F_2
$F_2 =$	
$F_3 =$	۱- تابع اصلی جدول صحت F_3 ۲- تابع ساده شده F_3
$F_3 =$	

A	B	C	D	F_3
۰	۰	۰	۰	۰
۰	۰	۰	۱	۰
۰	۰	۱	۰	۰
۰	۰	۱	۱	۰
۰	۱	۰	۰	۱
۰	۱	۰	۱	۱
۰	۱	۱	۰	۱
۰	۱	۱	۱	۰
۱	۰	۰	۰	۰
۱	۰	۰	۱	۰
۱	۰	۱	۰	۱
۱	۰	۱	۱	۰
۱	۱	۰	۰	۱
۱	۱	۰	۱	۰
۱	۱	۱	۰	۱
۱	۱	۱	۱	۱

(ج)

A	B	C	F_2
۰	۰	۰	۰
۰	۰	۱	۰
۰	۱	۰	۱
۰	۱	۱	۰
۱	۰	۰	۱
۱	۰	۱	۱
۱	۱	۰	۱
۱	۱	۱	۱

(ب)

A	B	C	F_1
۰	۰	۰	۰
۰	۰	۱	۰
۰	۱	۰	۱
۰	۱	۱	۰
۱	۰	۰	۱
۱	۰	۱	۱
۱	۱	۰	۰
۱	۱	۱	۱

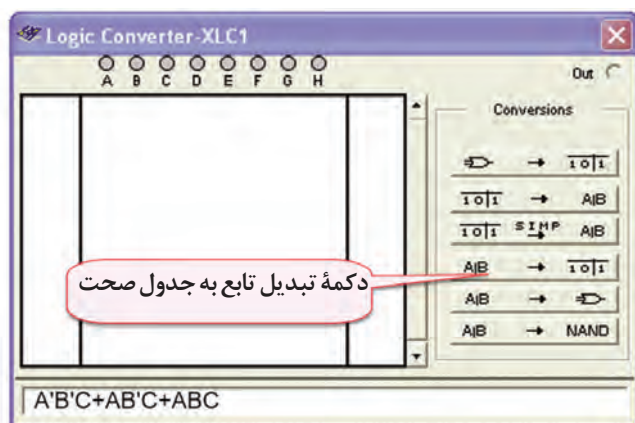
(الف)

شکل ۵۰

۴ به دست آوردن جدول صحت و پیاده سازی مدار تابع منطقی با گیت های منطقی یک عبارت بولی با دستگاه مبدل منطقی (Logic Converter)

۵ عبارت بولی (منطقی) $F_4 = \overline{A}\overline{B}C + A\overline{B}C + ABC$ را در محل تابع خروجی دستگاه مطابق شکل ۵۱ بنویسید.

- با فعال نمودن دکمه تبدیل تابع به جدول صحت، جدول صحت تابع به دست آمده F_4 را مشاهده کنید.
- سپس با توجه به جملات تابع F_4 یک‌های قسمت خروجی آن را تعیین کنید.
- با فعال کردن دکمه SIMP (simplify) شکل ساده شده عبارت منطقی را به دست آورده و یادداشت کنید.



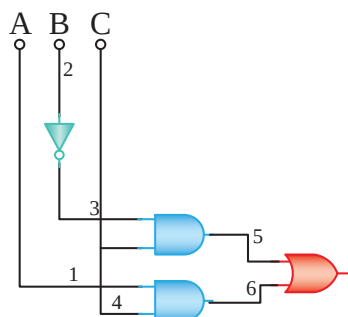
A	B	C	F_1
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

شکل ۵۱- کلید تبدیل تابع به جدول صحت

$F_4 =$

تابع ساده شده

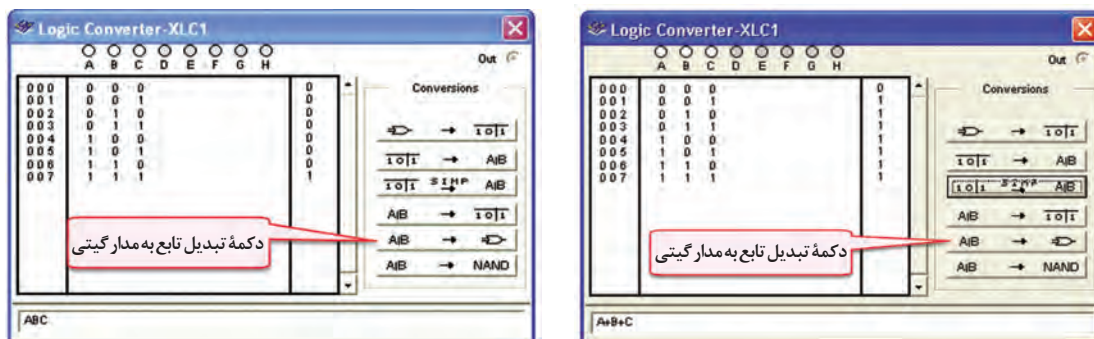
- در مرحله بعد دکمه تبدیل عبارت منطقی به گیت را انتخاب کنید، مدار پیاده‌سازی شده این تابع F_4 با گیت و عبارت ساده شده آن را در شکل ۵۲ مشاهده نمایید.



شکل ۵۲- کلید تبدیل تابع به مدار گیت‌های منطقی



بررسی کنید که دستگاه مبدل منطقی (Logic Converter) مدار گیت یک تابع OR یا AND سه ورودی را چگونه پیاده‌سازی می‌کند. مدار هریک را ترسیم کنید (شکل ۵۳).



شکل ۵۳- کلیدهای تبدیل تابع به مدار گیت‌های منطقی AND و OR سه ورودی

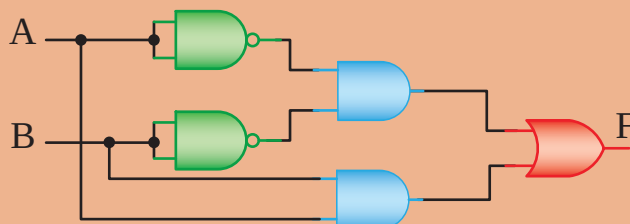
۵ مدار منطقی ساده شده هریک از توابع منطقی زیر را با دستگاه مبدل منطقی (Logic Converter) به دست آورید. نقشه مدار منطقی هریک از تابع‌های ساده شده در جدول ۲۹ را ترسیم کنید.

جدول ۲۹

تابع‌های منطقی	مدار منطقی تابع ساده شده	شماره IC‌های لازم برای پیاده‌سازی سخت‌افزاری
تابع اصلی F_5		
$F_5 = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$		
تابع ساده شده F_5		
$F_5 = \dots\dots\dots$		
تابع اصلی F_6		
$F_6 = \overline{A}\overline{B}C + A\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\overline{C} + \overline{A}B\overline{C} + A\overline{B}\overline{C}$		
تابع ساده شده F_6		
$F_6 = \dots\dots\dots$		
تابع اصلی F_7		
$F_7 = \overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD + A\overline{B}\overline{C}D + A\overline{B}CD + A\overline{B}\overline{C}D + A\overline{B}CD$		
تابع ساده شده F_7		
$F_7 = \dots\dots\dots$		



چگونه می‌توان رابطه منطقی (تابع) خروجی مدار شکل ۵۴ را با دستگاه مبدل منطقی (Logic Converter) به دست آورد، چه مدار ساده‌ای می‌توان جایگزین آن کرد؟ گزارش نتیجه تحقیق خود را به هنرآموز کارگاه ارائه دهید.



شکل ۵۴



ساخت پروژه پیشنهادی (۱)

هدف: کسب مهارت نوشتن تابع منطقی و ساده‌سازی آن، شبیه‌سازی و ساخت برد مدار دیجیتال
مواد، ابزار و تجهیزات: نوشت افزار، کاغذ، رایانه، نرم افزار مرتبط فیبر مدار چاپی (آلتیوم دیزاینر)، کاغذ گلاسه، اتو، چاپگر لیزر، قیچی، اسید پرکلرودوفر، لوازم اسیدکاری، ابزار و لوازم لحیم کاری، قطعات الکترونیکی مطابق با نقشه مدار منطقی پروژه انتخابی

مراحل اجرای پروژه:

جدول ۳۰- جدول صحت مدار

منطقی پروژه ۲

A	B	C	F _۱
۰	۰	۰	۰
۰	۰	۱	۰
۰	۱	۰	۰
۰	۱	۱	۱
۱	۰	۰	۰
۱	۰	۱	۱
۱	۱	۰	۱
۱	۱	۱	۱

۱- مدار دیجیتال مورد نظر را بررسی کنید.

«مدار منطقی را طرح کنید که خروجی آن تابع رأی اکثریت یک هیئت داورى سه نفرى باشد.»

برای بررسی رفتار یک مدار منطقی باید پاسخ مدار (تابع خروجی) را در همه حالات ورودی آن در نظر گرفت چون مدار دارای سه ورودی است و ۸ حالت متفاوت وجود دارد.

برای به دست آوردن تابع منطقی ابتدا باید جدول صحت آن را تشکیل داد. اگر سه عضو هیئت داورى با حروف A، B و C به عنوان ورودی‌ها و نتیجه آرای هیئت داورى را با F به عنوان خروجی در نظر بگیریم، براساس منطق حاکم بر رأی هیئت داورى هرگاه دو نفر از سه نفر و یا

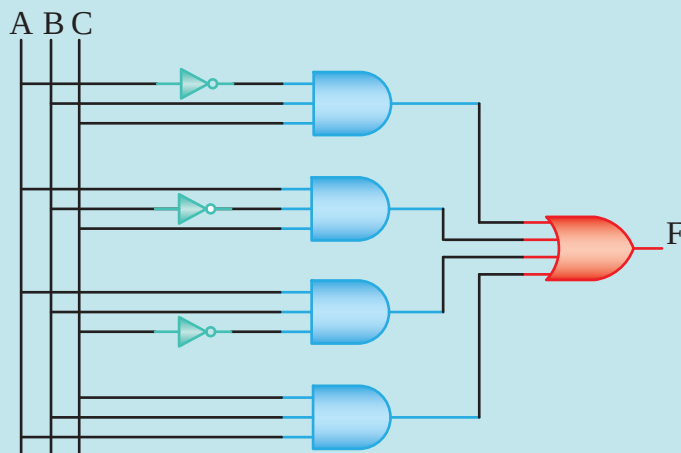
هرسه داور رأی تأیید دهند آن زمان چراغ سفید رأی هیئت داورى روشن خواهد شد. در واقع در این حالت خروجی مدار باید وجود داشته باشد (خروجی یک شود) و در سایر موارد رأی هیئت داورى مورد تأیید نیست (خروجی صفر باشد). (جدول ۳۰)

حال با در نظر گرفتن خواسته مدار، لازم است تا در ستون خروجی سطرهایی که رأی مثبت دو یا هر سه داور مثبت است به عنوان رأی اکثریت عدد ۱ و در سایر موارد عدد ۰ گذاشته شود.

با کمی دقت در جدول صحت مشاهده می‌شود، این مدار در ۴ حالت باید خروجی داشته باشد. به همین خاطر تابع منطقی آن را با توجه به وضعیت ورودی‌ها چنین می‌توان نوشت:

$$F = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$$

براساس تابع منطقی به دست آمده و بدون ساده سازی تابع شکل مدار منطقی به صورت شکل ۵۵ خواهد بود.



شکل ۵۵- مدار منطقی تابع F

۲ به روش جبر بول یا به کمک جدول کارنو یا با دستگاه مبدل منطقی (Logic Converter) صورت ساده شده تابع خروجی را می توان به دست آورد.

الف) با استفاده از قضایای جبر بول یا جدول کارنو می توان تابع منطقی اصلی را ساده کرد. در اینجا از جدول کارنو برای ساده سازی استفاده شده است. به همین دلیل ابتدا متناسب با هریک از جملات مینترم در خانه های جدول کارنو یک را می نویسیم (شکل ۵۶). سپس با استفاده قاعده تشکیل گروه می توان تابع ساده شده را به صورت شکل ۵۷ به دست آورد.

		A					
		AB		00	01	11	10
C	0			1			
	1		1	1	1		
				B			



		A					
		AB		00	01	11	10
C	0			1			
	1	1	1	1			
				B			

$F = BC + AC + AB$

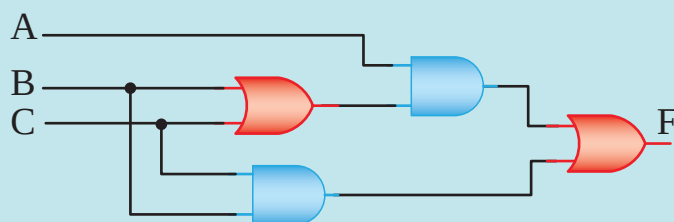
شکل ۵۶- جدول صحت تابع F

شکل ۵۷- شکل ساده شده تابع F

برای اتصال مدار منطقی تابع ساده شده با جدول کارنو نیاز به سه گیت AND و دو گیت OR است. با توجه به اینکه جملات تابع ساده شده در برخی حروف اشتراک دارند لذا با بهره گیری از قاعده فاکتورگیری می توان شکل تابع را خلاصه تر و تعداد گیت های مورد نیاز را کم کرد.

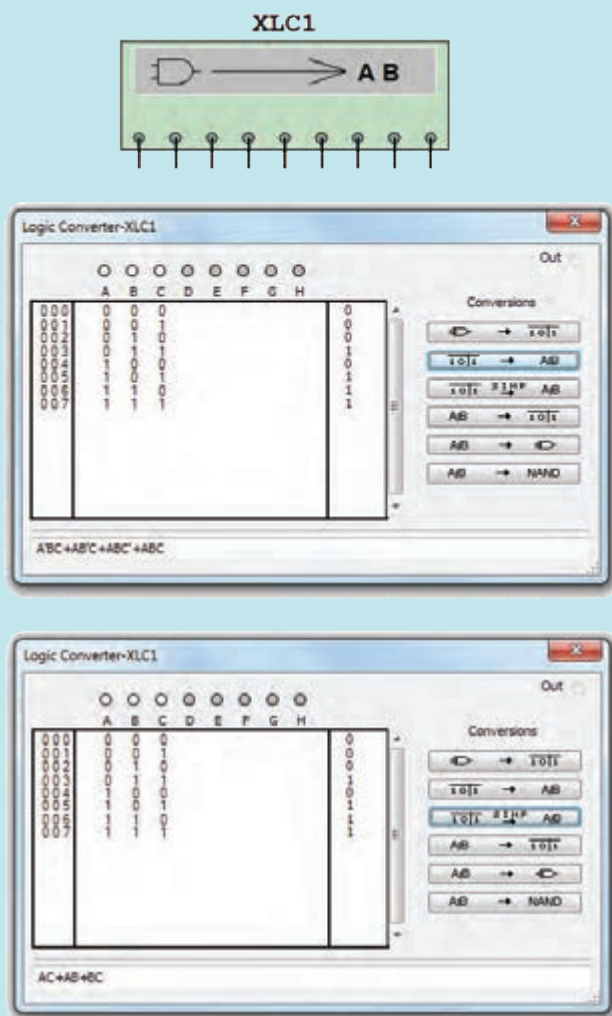
شکل خلاصه شده تابع $F = BC + A \cdot (B+C)$

شکل ۵۸ مدار منطقی عبارت ساده شده نهایی را نشان می‌دهد.



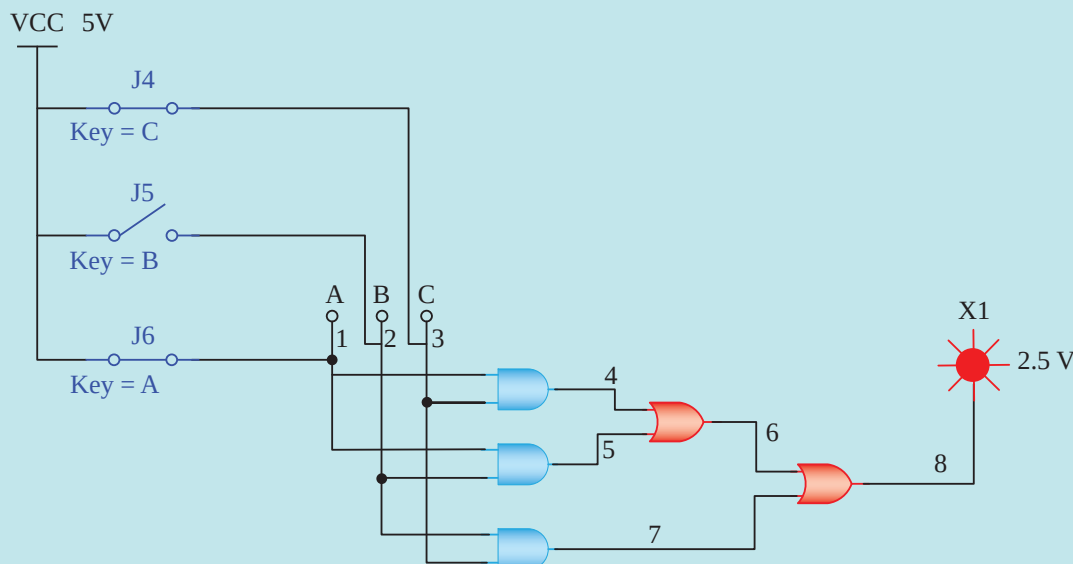
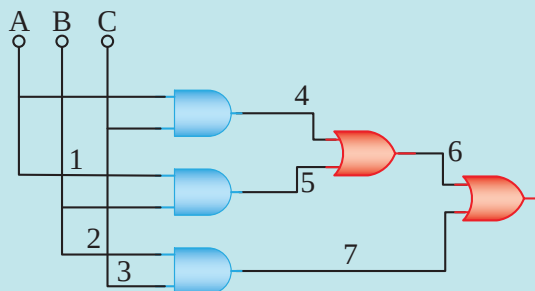
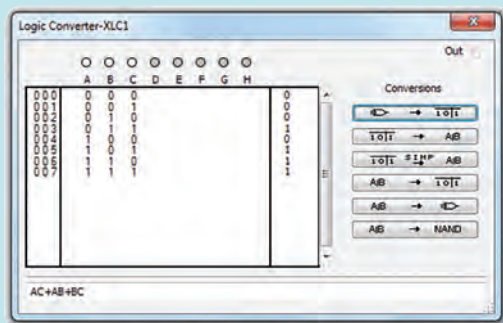
شکل ۵۸

ب) با دستگاه مبدل منطقی (Logic Converter) می‌توانید صورت ساده شده تابع خروجی را به‌دست آورید.



شکل ۵۹- تابع ساده شده مدار منطقی پروژه ۱ با دستگاه مبدل منطقی (Logic Converter)

۳ مدار منطقی تابع ساده شده را با دستگاه مبدل منطقی (Logic Converter) به دست آورید و آن را در محیط نرم افزار شبیه سازی کنید تا از صحت عبارت منطقی به دست آمده اطمینان حاصل کنید.



شکل ۶۰- پیاده سازی و اجرای مدار منطقی با دستگاه مبدل منطقی (Logic Converter)

۴ فهرست قطعات مورد نیاز را تهیه کنید.

۵ مدار منطقی پروژه انتخاب شده را بر روی بردبرد بندید و آن را آزمایش کنید.

فعالیت عملی: قابل توجه هنرجویان علاقه مند

در صورتی که وقت لازم برای انجام ادامه کار عملی را داشتید مراحل ۶ تا ۱۰ را نیز انجام دهید.

۶ نقشه الکترونیکی مدار را به محیط نرم افزار آلتیوم دیزاینر انتقال دهید سپس با رعایت استانداردها و نکات طراحی، نقشه PCB آن را رسم کنید.

۷ فیبر مدار چاپی (اندازه فیبر مدار چاپی با پیشنهاد مربی کارگاه تعیین شود) تهیه شود.

۸ تمامی قطعات بر روی برد نصب و لحیم کاری پایه های آنها صورت گیرد.



۹ با تنظیم مقدار ولتاژ تغذیه مدار، آن را به ورودی مدار اتصال دهید و خروجی مدار منطقی را با آزمایش مشاهده کنید.

۱۰ گزارش کار انجام پروژه را به صورت فایل رایانه‌ای مستندسازی کنید و آن را برای ارزشیابی به هنرآموز خود ارائه دهید.

ساخت پروژه پیشنهادی (۲)

هدف: کسب مهارت نوشتن تابع منطقی و ساده‌سازی آن، شبیه‌سازی و ساخت برد مدار دیجیتال

مواد، ابزار و تجهیزات: نوشت افزار، کاغذ، رایانه، نرم افزار مرتبط فیبر مدار چاپی (آلتیوم دیزاینر) کاغذ گلاسه، اتو، چاپگر لیزر، قیچی، اسید پرکلرودوفر، لوازم اسیدکاری، ابزار و لوازم لحیم کاری، قطعات الکترونیکی مطابق با نقشه مدار منطقی پروژه انتخابی

حسگر زمین لرزه $A = S \rightarrow$
 حسگر نشت گاز $B = G \rightarrow$
 حسگر نشت آب $C = W \rightarrow$
 حسگر حریق ساختمان $D = F \rightarrow$

۱ مدار دیجیتالی را با توجه به توضیحات داده شده زیر طراحی کنید.

مداری به عنوان سامانه هشداردهنده حفاظت ساختمان در مقابل زلزله به گونه‌ای طراحی کنید که عملکرد آن به صورت زیر باشد.

■ در صورت بروز زمین لرزه (با تشخیص حسگر S) که موجب تحریک یکی از حسگرهای نشت گاز G، نشت آب W و حریق، ساختمان F یا ترکیب آنها فعال شوند یک لامپ سیگنال قرمز روشن و آلام صوتی به صدا درآید.

■ در غیر این صورت یک چراغ سبز روشن باشد.

جدول ۳۱

ردیف	A	B	C	D	Y
۰	۰	۰	۰	۰	۰
۱	۰	۰	۰	۱	۰
۲	۰	۰	۱	۰	۰
۳	۰	۰	۱	۱	۰
۴	۰	۱	۰	۰	۰
۵	۰	۱	۰	۱	۰
۶	۰	۱	۱	۰	۰
۷	۰	۱	۱	۱	۰
۸	۱	۰	۰	۰	۰
۹	۱	۰	۰	۱	۱
۱۰	۱	۰	۱	۰	۱
۱۱	۱	۰	۱	۱	۱
۱۲	۱	۱	۰	۰	۱
۱۳	۱	۱	۰	۱	۱
۱۴	۱	۱	۱	۰	۱
۱۵	۱	۱	۱	۱	۱

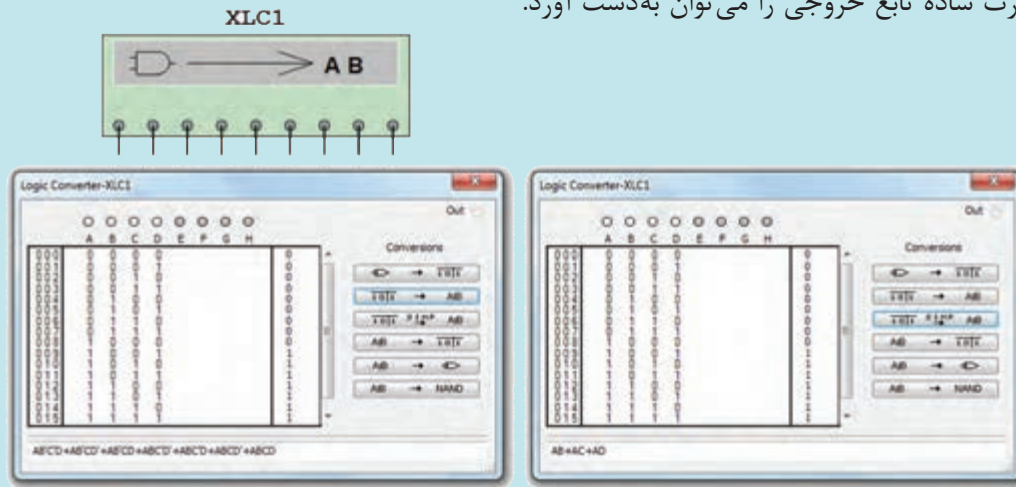
۲ برای بررسی رفتار یک مدار منطقی باید پاسخ مدار (تابع خروجی) را برای همه حالات ورودی آن در نظر گرفت چون مدار دارای چهار ورودی است، ۲^۴ حالت متفاوت وجود دارد. برای به دست آوردن تابع منطقی ابتدا باید جدول صحت آن را تشکیل داد (جدول ۳۱).

توضیح



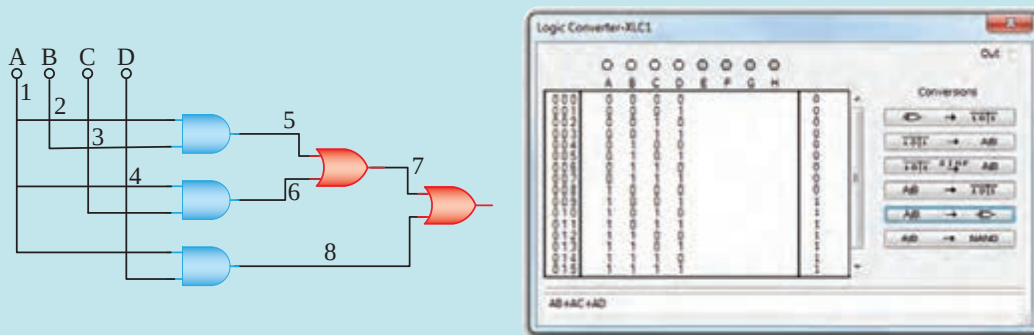
در جدول صحت از حروف A تا D به عنوان مشخصه سنسورهای موردنظر مدار استفاده شده است.

۳ سپس به کمک قضایای جبر بول یا جدول کارنو یا با دستگاه مبدل منطقی (Logic Converter) صورت ساده تابع خروجی را می توان به دست آورد.



شکل ۶۱- پیاده سازی و اجرای مدار منطقی پروژه ۲ با دستگاه مبدل منطقی (Logic Converter)

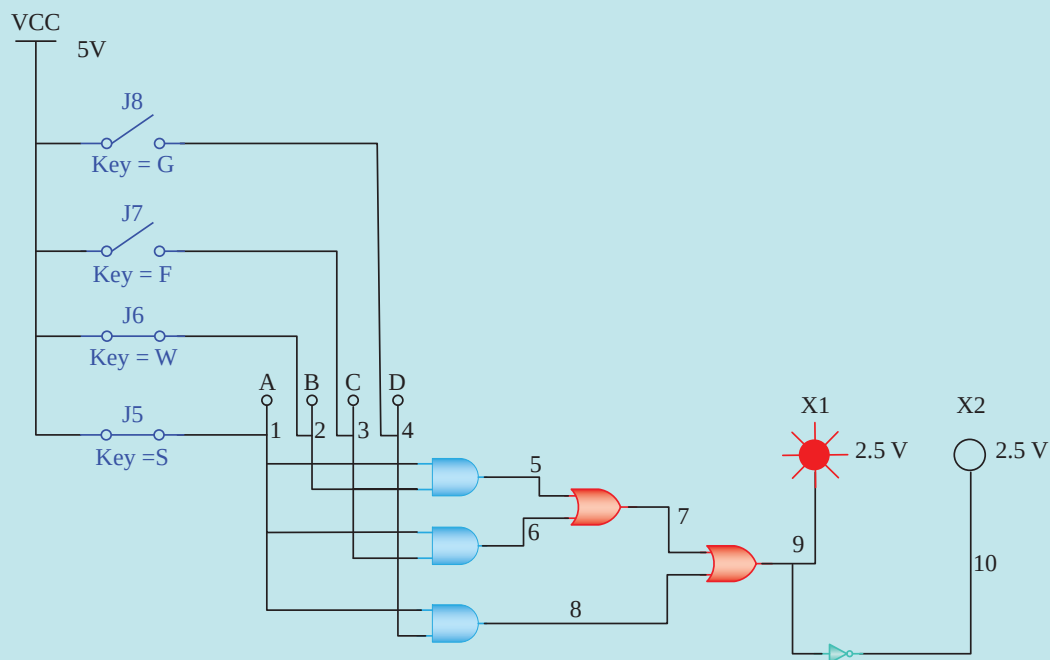
۴ مدار منطقی آن را با دستگاه مبدل منطقی (Logic Converter) به دست آورید و آن را در محیط نرم افزار شبیه سازی کنید تا از صحت عبارت منطقی به دست آمده اطمینان حاصل کنید.



نکته مهم



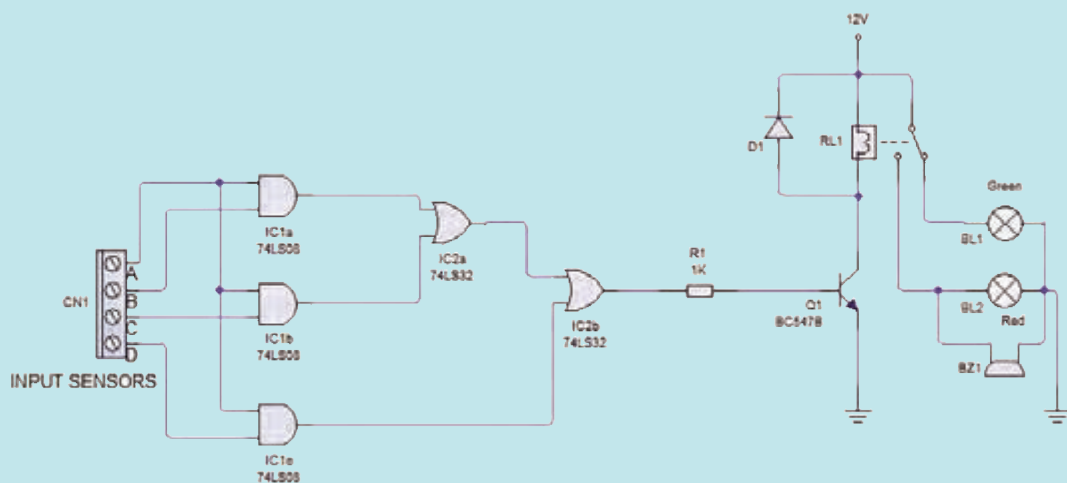
با به کارگیری قوانین جبر بول آیا می توان تعداد گیت های AND را کاهش داد؟ مدار طرح کنید.



شکل ۶۲- پیاده‌سازی و اجرای مدار منطقی پروژه ۲ با دستگاه مبدل منطقی (Logic Converter)

۵ فهرست قطعات موردنیاز مدار الکترونیکی پروژه انتخاب شده شکل ۶۳ را تهیه کنید

۶ مدار پروژه انتخاب شده را بر روی بردبرد ببندید و آن را آزمایش کنید.



شکل ۶۳- مدار الکترونیکی پروژه سامانه هشداردهنده حفاظت ساختمان

۷ نقشه الکترونیکی مدار را به محیط نرم‌افزار آلتیوم دیزاینر انتقال دهید سپس با رعایت استانداردها و نکات طراحی، نقشه PCB آن را رسم کنید.

۸ فیبر مدار چاپی (اندازه فیبر مدار چاپی با پیشنهاد مربی کارگاه تعیین شود) تهیه شود.

۹ تمامی قطعات بر روی برد نصب و لحیم کاری پایه‌های آنها صورت گیرد.

۱۰ با تنظیم مقدار ولتاژ تغذیه مدار، آن را به ورودی مدار اتصال دهید و خروجی مدار منطقی را با آزمایش مشاهده کنید.

۱۱ گزارش کار انجام پروژه را به صورت فایل رایانه‌ای مستندسازی کنید و آن را برای ارزشیابی به هنرآموز خود ارائه دهید.

خود ارزیابی

۱ تابع منطقی گیت مقابل کدام گزینه است؟

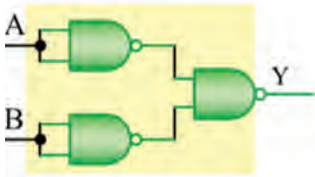


(۱) $AB + \overline{A}\overline{B}$ (۲) $\overline{A}B + A\overline{B}$ (۳) $\overline{A}B + AB$ (۴) $\overline{A}\overline{B} + \overline{A}B$

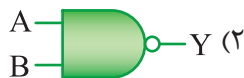
۲ عضو خنثی در گیت OR و در گیت AND است.

(به ترتیب از راست به چپ)

(۱) ۰-۰ (۲) ۱-۰ (۳) ۱-۱ (۴) ۰-۱



۳ گیت معادل شکل مقابل کدام گزینه است؟



A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

۴ جدول درستی مربوط به کدام گیت است؟



$F = \overline{A}\overline{B} + A\overline{B} + \overline{A}B$

۵ کدام گزینه ساده‌ترین فرم تابع F را بیان می‌کند؟

(۱) $\overline{A}$ (۲) $\overline{B}$ (۳) $A\overline{B}$ (۴) $\overline{A} + \overline{B}$

	A	
	1	1
B	0	1

۶ ساده‌ترین تابع منطقی نقشه کارنو کدام گزینه است؟

$$A + \overline{B} \quad (۴)$$

$$\overline{B} + AB \quad (۳)$$

$$\overline{A} + AB \quad (۲)$$

$$A + \overline{A}\overline{B} \quad (۱)$$

		A		
	1	0	1	1
C	1	0	1	1
	B			

۷ ساده‌ترین تابع منطقی نقشه کارنو کدام گزینه است؟

$$A + \overline{B} \quad (۴)$$

$$\overline{A} + AB \quad (۳)$$

$$A + B \quad (۲)$$

$$A + \overline{A}\overline{B} \quad (۱)$$

ارزشیابی شایستگی اجرای مدارهای منطقی

نام و نام خانوادگی		شماره ملی	تاریخ ارزشیابی:	نام
کد حرفه	۸۲۱۲۱۲	حرفه :	مونتاز کاران تجهیزات و وسایل برقی و الکترونیکی	کد پیمان/پودمان
کد وظیفه شغل	۸۲۱۲۱۲۰۳	وظیفه شغل:	ساخت مدارهای دیجیتال و میکروکنترلر	سطح صلاحیت
کد کار	۸۲۱۲۱۲۰۳۲۲	کار	اجرای مدار عملگرهای منطقی	سطح شایستگی

ردیف	مراحل کار	شرایط انجام کار (ابزار، مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره دهی)	نتایج ممکن	نمره کسب شده			
۱	ساده‌سازی توابع منطقی با استفاده از قوانین جبر بول	برگه تمرین توابع منطقی، قلم و کاغذ یا نرم‌افزار آموزشی، قوانین جبر بول و زمان ۲۰-۱۵ دقیقه	بررسی پاسخ‌ها و مقایسه با پاسخ صحیح، چک‌لیست مراحل ساده‌سازی و پرسش شفاهی	ساده‌سازی کامل و صحیح	۳				
					ساده‌سازی با خطاهای جزئی و نتیجه قابل قبول		۲		
					خطا در قوانین و نتیجه نادرست		۱		
۲	ساده‌سازی توابع منطقی با استفاده از نقشه کارنو و تشخیص مینترم و ماکسترم	برگه تمرین توابع منطقی، قلم و کاغذ، الگوی نقشه کارنو برای ۳ یا ۴ متغیر و زمان حدود ۲۰-۱۵ دقیقه	بررسی نقشه کارنو، کنترل گروه‌بندی، مقایسه نتیجه با پاسخ صحیح و چک‌لیست ارزیابی	تکمیل صحیح کارنو و ساده‌سازی صحیح خطای جزئی در گروه‌بندی یا عبارت نهایی	۳				
					انتقال یا گروه‌بندی نادرست و نتیجه اشتباه		۱		
					۲				
۳	درک و تفسیر نتایج خروجی نرم‌افزارهای طراحی مدار منطقی	رایانه مجهز به نرم‌افزار طراحی و شبیه‌سازی منطقی، فایل تمرین مشخص، دفترچه یادداشت و زمان ۲۵-۲۰ دقیقه در سایت رایانه	مشاهده عملکرد در نرم‌افزار، بررسی پاسخ‌های تحلیلی، چک‌لیست و پرسش شفاهی	تفسیر صحیح نتایج نرم‌افزار	۳				
					تفسیر قابل قبول با خطاهای جزئی		۲		
					تفسیر نادرست یا عدم درک خروجی‌ها		۱		
۴	ساخت و تست مدار منطقی ساده‌شده با گیت‌های AND، OR، NOT و اجرای یک پروژه عملی	برد برد یا برد آموزشی، آی‌سی گیت‌های منطقی، منبع تغذیه، سیم‌های رابط، LED و مقاومت، طرح مدار ساده‌شده و زمان ۴۰-۳۰ دقیقه در کارگاه الکترونیک	مشاهده عملکرد مدار، تطابق خروجی با جدول درستی، چک‌لیست اتصالات و پرسش شفاهی	ساخت صحیح و اجرای کامل مدار	۳				
					اجرای مدار با نقص‌های جزئی و عملکرد قابل قبول		۲		
					مدار نادرست یا عدم عملکرد صحیح پروژه		۱		
	شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش:	آزمایشگاه مدارهای منطقی، تجهیزات آموزشی و راهنمای کار، تجهیزات کارگاهی و حفاظت فردی و دستورالعمل‌های ایمنی و بهداشت حرفه‌ای	مشاهده رفتار هنرجو در حین انجام فعالیت، میزان رعایت اصول ایمنی و نظم، فعالیت‌های گروهی، مدیریت زمان و مسئولیت‌پذیری	رعایت کامل ایمنی، بهداشت و محیط‌زیست، رفتار حرفه‌ای، نظم و مسئولیت‌پذیری بالا	۲				
					رعایت نسبی ایمنی و بهداشت با خطاهای جزئی، رفتار کاری مناسب		۱		
ارزشیابی کار (شایستگی انجام کار)									
☐ بلی									
☐ خیر									

توضیحات:

۷- معیار شایستگی انجام کار:

۸- کسب حداقل نمره ۲ در مراحل بحرانی شامل اعمال صحیح قوانین جبر بول، ساده‌سازی تابع با نقشه کارنو، تحلیل درست خروجی نرم‌افزار و پیاده‌سازی عملی مدار ساده‌شده و اجرای پروژه ساده مدار منطقی است. (مراحل بحرانی).

- کسب حداقل نمره ۲ از بخش شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

- کسب حداقل میانگین ۲ از مراحل کار

۹- تعریف سطوح شایستگی: صرف نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

- مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

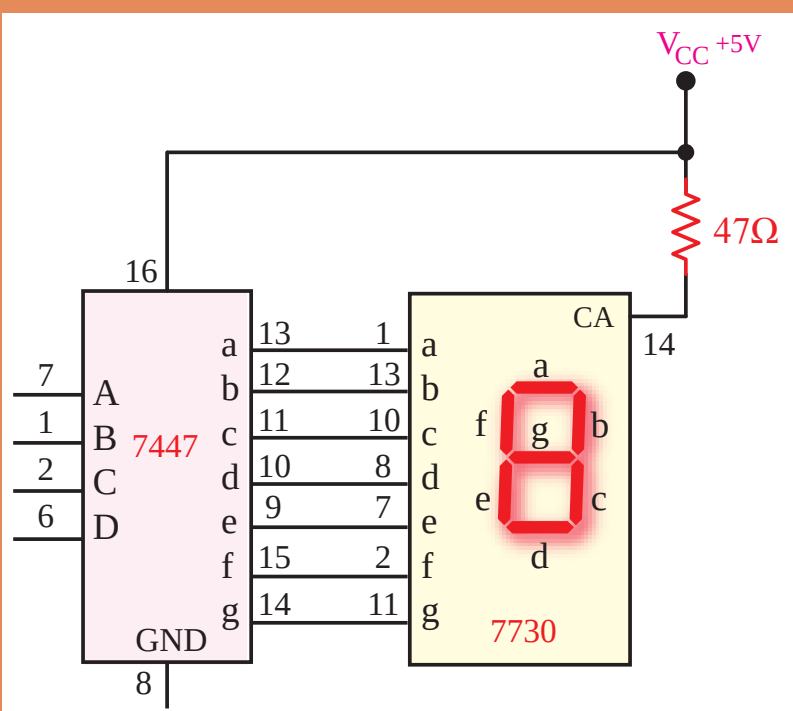
- تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.





پودمان سوم

پیاده‌سازی مدارهای کاربردی دیجیتال



واحد یادگیری ۱: پیاده‌سازی مدارهای ترکیبی

آیا تا به حال فکر کرده‌اید

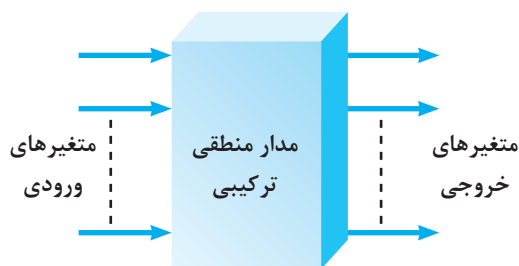
- طراحی مدارهای منطقی پیشرفته براساس یک اصول واحد صورت می‌گیرد.
 - چرا در طراحی مدارهای دیجیتال از تراشه‌های (IC) آماده استفاده می‌شود؟
 - در ماشین حساب محاسبات ریاضی مانند عمل جمع و تفریق چگونه انجام می‌شود؟
 - برای آدرس‌دهی، جهت فرمان دادن به دستگاه‌ها چه نوع مداری به کار می‌رود؟
 - کدام مدار دیجیتالی با انتخاب یکی از چند ورودی، باعث کاهش تعداد سیم‌ها و افزایش کارایی مدار می‌شوند؟
 - چگونه اطلاعات ورودی مانند فشردن کلیدهای صفحه‌کلید را به کدهای باینری قابل پردازش برای مدارهای دیجیتال تبدیل می‌کنند؟
- برای درک عملکرد مدارهای کاربردی دیجیتال، نیاز به تحلیل مدارهای ترکیبی داریم. در این مدارها، خروجی‌ها براساس متغیرهای ورودی و طبق تابع منطقی داده‌شده به‌دست می‌آیند. طراحی هر نوع مدار منطقی که خروجی‌های آن به‌طور هم‌زمان به تغییرات ورودی‌ها وابسته باشد، از اصول طراحی مدارهای ترکیبی پیروی می‌کند. در این واحد یادگیری، به تشریح این گونه مدارها می‌پردازیم.
- مدارهایی مانند مدارهای محاسبه‌گر (برای انجام عملیات ریاضی)، نمایش ارقام بر روی نمایشگرها (رمزگشاها)، مبدل صفحه‌کلید به ارقام باینری (رمزگذار)، مالتی‌پلکسرها (برای انتخاب و هدایت یکی از چند ورودی به یک خروجی) و مقایسه‌گرها (برای مقایسه مقادیر دودویی و تعیین بزرگ‌تر، کوچک‌تر یا مساوی بودن آنها) از مدارهای ترکیبی خاص به‌شمار می‌آیند.
- بدیهی است برای تحلیل، بررسی و اجرای عملی یک مدار کاربردی، رعایت نکات ایمنی و بهداشتی در تمامی مراحل کار ضروری است. همچنین توجه به مهارت‌های غیرفنی مانند کار گروهی، مسئولیت‌پذیری و پایبندی به نظم و ترتیب از مواردی است که اهمیت ویژه‌ای دارد و باید رعایت شود.

استاندارد عملکرد

طراحی، ساخت و پیاده‌سازی مدارهای منطقی ترکیبی شامل جمع‌گر، مقایسه‌گر، مالتی پلکسر، رمزگشا و رمزگذار

مدارهای ترکیبی

مدارهای ترکیبی، مدارهایی هستند که خروجی‌های آن به‌طور هم‌زمان به ورودی‌های آن وابسته است. مدار ترکیبی شامل متغیرهای ورودی، دروازه‌های منطقی و متغیرهای خروجی است. شکل ۱ بلوک دیاگرام مدار ترکیبی را نشان می‌دهد.



شکل ۱- بلوک دیاگرام مدار ترکیبی

اصول طراحی مدارهای ترکیبی

برای طراحی مدارهای منطقی ترکیبی مراحل زیر به ترتیب انجام می‌شود.
توجه: هدف از بیان این قسمت، آموزش طراحی مدار دیجیتال نیست، بلکه هدف آشنایی کلی با فرایند و مراحل طراحی است. هنرجویان علاقه‌مند به طراحی مدارهای دیجیتالی می‌توانند به منابع متعددی که در این زمینه وجود دارد مراجعه کنند.

■ تحلیل مسئله تعریف شده و تعیین تعداد ورودی و خروجی موردنیاز و در نهایت رسم بلوک دیاگرام.

■ تشکیل جدول صحت و نوشتن تابع منطقی مدار و ساده‌سازی آن.

در این قسمت بعضی از مدارهای ترکیبی با کاربردهای ویژه، که مصرف عام دارند و به‌صورت تراشه‌های (IC) تجارتي عرضه می‌شوند را معرفی می‌کنیم. در هر مورد، اصول کار مدار را با توجه به مفاهیم اساسی جبر بول بیان می‌کنیم، سپس بحث را با معرفی تراشه‌های تجارتي به پایان می‌رسانیم. علت انتخاب این روش نگاه سیستمی به مدار است.

مثال



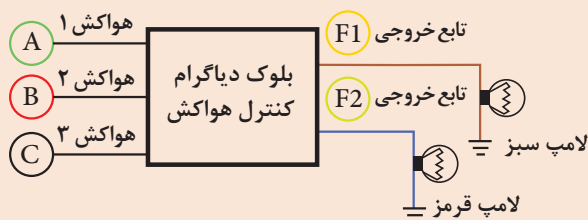
در یک پارکینگ، از سه هواکش جهت تهویه هوا استفاده شده است، که به شرح زیر عمل می‌کنند:

■ هنگامی که حداقل دو هواکش کار می‌کند، یک لامپ سبز روشن می‌شود.

■ در سایر حالات یک لامپ قرمز روشن می‌شود.

■ مدار منطقی کنترل این هواکش‌ها را طراحی کنید.

حل: با توجه به خواسته‌های مسئله، بلوک دیاگرام مدار را مطابق شکل ۱ رسم می‌کنیم.



شکل ۲- بلوک دیاگرام مثال

با توجه به بلوک دیاگرام شکل ۲ جدول صحت مربوط به عملکرد هواکش‌ها را رسم می‌کنیم. این جدول صحت، سه ورودی A، B و C و خروجی F_1 و F_2 را مطابق جدول ۱ خواهد داشت.

تابع F_1 و F_2 را به ترتیب از جدول صحت استخراج می‌کنیم.

جدول ۱- جدول صحت مدار مثال صفحه قبل

A	B	C	F_1	F_2
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

$$F_1 = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$F_2 = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C}$$

با توجه به جدول در این مسئله همواره برای خروجی‌ها، رابطه $F_2 = \bar{F}_1$ برقرار است. تابع F_1 و F_2 را با استفاده نقشه کارنو ساده می‌کنیم. همچنین می‌توانیم برای مثال F_1 یا F_2 را به دست آوریم، سپس آن را NOT می‌کنیم تا دیگری به دست آید. اگر تابع F_1 را NOT کنیم تابع F_2 حاصل می‌شود. بنابراین کافی است که ابتدا یکی از توابع F_1 یا F_2 را محاسبه کنیم، سپس با NOT کردن تابع به دست آمده تابع دومی را تعیین کنیم. در این مرحله ابتدا F_1 را به دست می‌آوریم، سپس آن را NOT می‌کنیم تا F_2 مشخص شود. حال تابع F_1 را با نقشه کارنو جدول ۲ ساده می‌کنیم.

جدول ۲- نقشه کارنو تابع F_1

$$F_1 = \bar{A}BC + A\bar{B}C + AB\bar{C} + ABC$$

$$F_1(A,B,C) = \sum(3,5,6,7)$$

AB		00	01	11	10
C	0			1	
	1		1	1	1

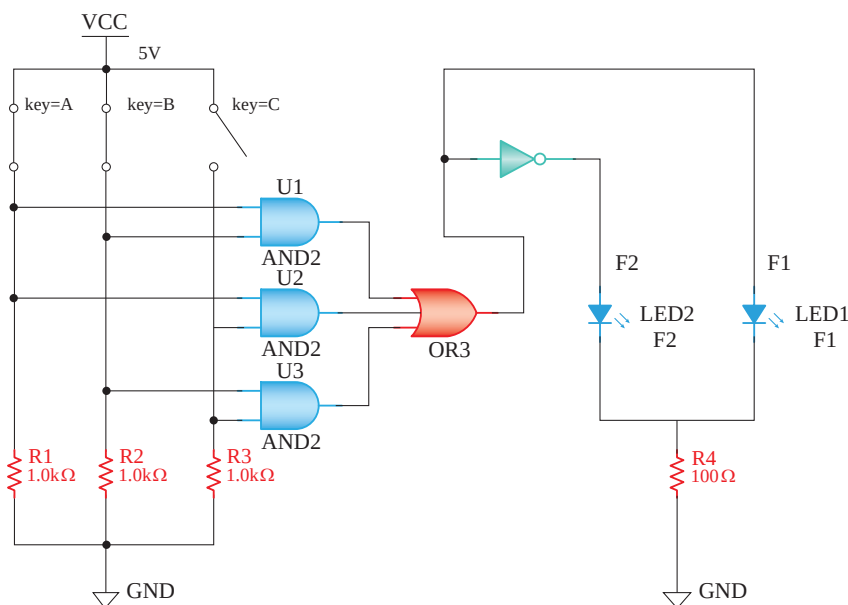
تابع F_1 ساده شده از نقشه کارنو برابر است با:

$$F_1 = AB + AC + BC$$

بر همین اساس تابع ساده شده F_2 را چنین می‌توان نوشت :

$$F_2 = \bar{F}_1 = \overline{(AB + AC + BC)}$$

حال تابع F_1 و F_2 را مطابق شکل ۳ توسط گیت‌های منطقی پایه پیاده‌سازی می‌کنیم.



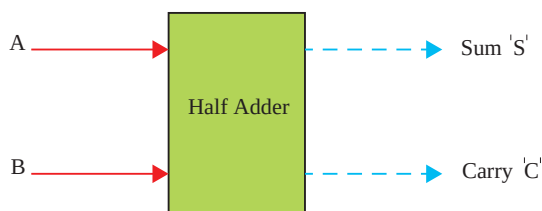
شکل ۳- مدار منطقی کنترل هوا

مدارهای ترکیبی پایه

مدارهای ترکیبی پایه در انواع مختلف ساخته می‌شوند که در اینجا به ترتیب به بررسی آنها می‌پردازیم.

۱ مدارهای جمع‌کننده

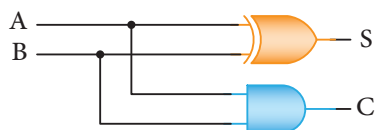
الف) جمع‌کننده ناقص (نیم جمع‌کننده H.A - Half Adder): جمع دو رقم دودویی را می‌توان با مداری به نام جمع‌کننده ناقص یا به اختصار H.A انجام داد. (شکل ۵) مدار H.A مداری است که دو ورودی (A و B) و دو خروجی (S و C) دارد، خروجی‌ها مجموع ارقام ورودی‌ها هستند. در خروجی رقم اول حاصل جمع را با S (SUM) و رقم نقلی را با C (Carry) مشخص می‌کنند. جدول صحت جمع دو بیت A و B را در جدول ۳ مشاهده می‌کنید.



شکل ۴- بلوک دیاگرام جمع‌کننده ناقص

جدول ۳- جدول صحت H.A

A	B	C	S	
0	0	0	0	$0 + 0 = 0$
0	1	0	1	$0 + 1 = 1$
1	0	0	1	$1 + 0 = 1$
1	1	1	0	$1 + 1 = 10$



شکل ۵- مدار جمع‌کننده ناقص

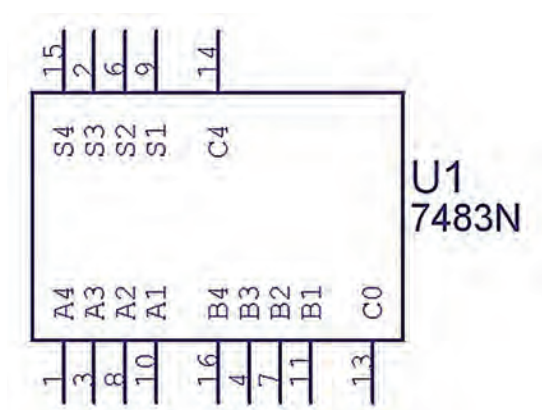
مدار منطقی نیم جمع‌کننده مطابق شکل ۵ است.

تحقیق کنید



رقم نقلی چیست؟ با مراجعه به منابع مختلف مفهوم آن را بیابید و نتیجه را در قالب یک گزارش ارائه کنید.

ب) جمع کننده کامل (تمام جمع کننده Full Adder - H.A): برای جمع دو عدد باینری چند رقمی از مدار تمام جمع کننده که متشکل از چند مدار نیم جمع کننده است باید استفاده کرد. برای اتصال این مدار از گیت های منطقی یا IC خاص می توان استفاده کرد. IC 7483 که در شکل ۶ نشان داده شده یک جمع گر چهار بیتی است. برای جمع کردن دو عدد چهار بیتی $A(A_4A_3A_2A_1)$ و $B(B_4B_3B_2B_1)$ از آن می توان استفاده کرد.



شکل ۶- IC جمع کننده ۴ بیتی

تحقیق کنید



تراشه (IC) ۷۴۸۳ از چند Full Adder تشکیل شده است؟

کار عملی ۱



بستن مدارهای جمع کننده به صورت نرم افزاری و سخت افزاری

هدف: کسب مهارت در بررسی عملکرد مدارهای ترکیبی خاص (جمع گر)

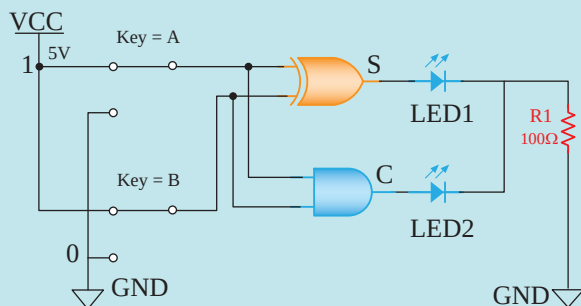
مواد، ابزار و تجهیزات: لوازم التحریر - اینترنت - رایانه - نرم افزار مولتی سیم یا هر نرم افزار مناسب دیگر - دیود نورانی LED قرمز و سبز - مقاومت های 100Ω ، $\frac{1}{4}$ وات - بردبرد - سیم های بردبرد - میز آزمایشگاهی الکترونیک با تجهیزات استاندارد - کلید یک پل کوچک (مینیا توری، قابل نصب روی بردبرد)

- برگه اطلاعات آی سی های دیجیتال - IC ۷۴۸۳

مراحل اجرای کار:

۱ مدار جمع کننده ناقص را مطابق شکل ۷ در محیط نرم افزار ببندید.

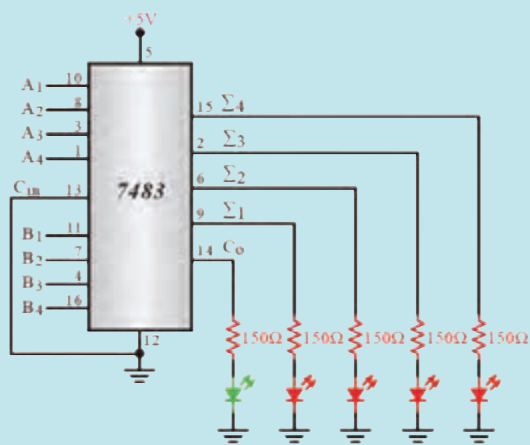
۲ با تغییر وضعیت کلیدها در مدار شکل ۷، جدول ۴ صحت مدار جمع‌کننده ناقص را کامل کنید.



شکل ۷- مدار عملی جمع‌کننده ناقص

جدول ۴- جدول صحت مدار جمع‌کننده ناقص

A	B	Co	S
0	0		
0	1		
1	0		
1	1		



شکل ۸

۲ در کدام حالت از ورودی‌ها، بیت نقلی ظاهر شده است؟

۴ مدار شکل ۸ یک جمع‌کننده ۴ بیتی با IC۷۴۸۳ است. مدار را روی بردبرد ببندید.

۵ با انتخاب مقادیر مختلف برای دو عدد باینری چهار بیتی $A=A_4A_3A_2A_1$ و $B=B_4B_3B_2B_1$ وضعیت خروجی جمع‌کننده را مشاهده کنید. وضعیت روشن شدن دیودهای نورانی، عدد باینری خروجی را نشان می‌دهد.

می‌توانید از جدول ۵ کمک بگیرید.

جدول ۵- جدول جمع‌کننده ۴ بیتی

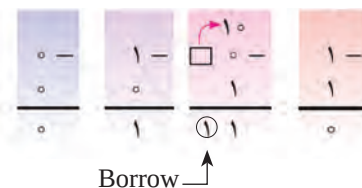
وضعیت بیت‌های عدد A					وضعیت بیت‌های عدد B					وضعیت بیت‌های خروجی					عدد حاصل
عدد A	A4	A3	A2	A1	عدد B	B4	B3	B2	B1	Co	S4	S3	S2	S1	A + B
3	0	0	1	1	2	0	0	1	0	0	0	1	0	1	5
	1	0	0	1		0	1	1	0						
	1	1	0	0		0	1	0	1						
	1	0	0	1		0	0	1	1						

۲ مدارهای تفریق کننده

مدارهای تفریق را نیز به دو شکل می توان اتصال داد.

■ **تفریق کننده ناقص (نیم تفریق کننده H.S-Half Subtractor)** برای تفریق دو رقم باینری از مدار تفریق کننده ناقص (H.S) براساس منطق زیر استفاده می شود.

تفریق یک عدد باینری از عدد باینری دیگر با روشی مانند تفریق اعداد دسیمال انجام می پذیرد، یعنی اگر رقم بزرگ تر از رقم کوچک تر کم شود یک واحد از مکان بعدی قرض گرفته می شود و در مکانی که یک واحد قرض گرفته شده یک را به صفر تبدیل می کنیم. واحد قرض گرفته شده Borrow می گویند.

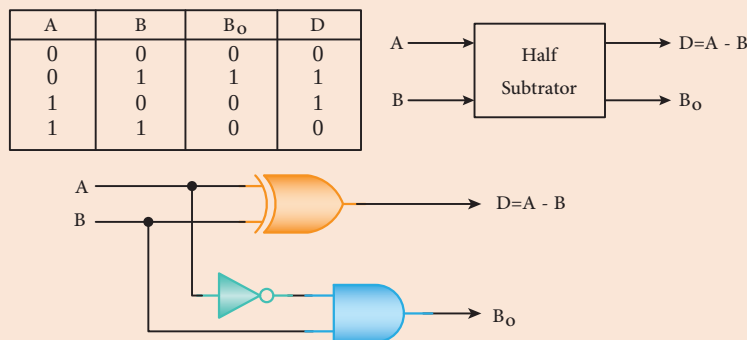


در تفریق $\frac{1}{2}$ یک واحد Borrow ایجاد شد، یک به مکان کم ارزش تر انتقال می یابد و ۱۰ می شود.



شکل ۹- عملیات تفریق

با بهره گیری از منابع مختلف و کمک معلم خود پیرامون عمل تفریق دو عدد تک بیتی (A-B) شکل ۱۰ تحقیق کنید.



شکل ۱۰- جدول صحت، بلوک دیاگرام و مدار تفریق گر ناقص دو عدد تک بیتی

آیا می توانید با یک گیت XOR و یک گیت AND تفریق کننده تک بیتی را طراحی کنید.

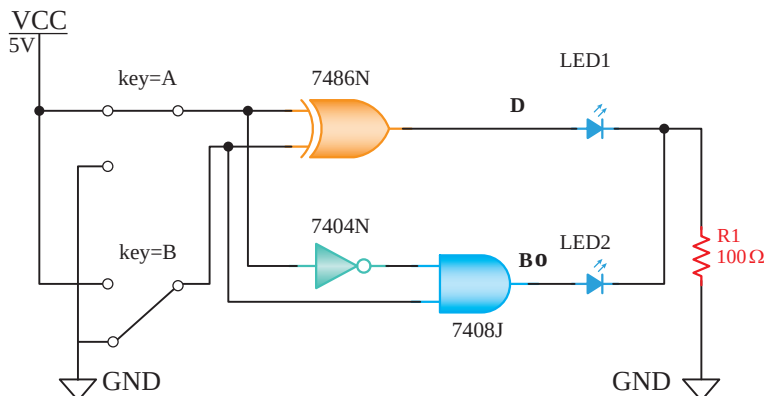
فعالیت



تحقیق کنید



مدار تفریق گر ناقص را مطابق شکل ۱۱ در نرم‌افزار مولتی سیم پیاده‌سازی کنید.



شکل ۱۱- مدار عملی تفریق‌کننده ناقص (تفریق دو عدد تک بیتی)

جدول ۶- جدول صحت
مدار تفریق گر ناقص

A	B	B ₀	D
0	0		
0	1		
1	0		
1	1		

کلیدهای ورودی A و B مدار شکل ۱۰ را مطابق جدول ۶ تغییر وضعیت دهید و خروجی‌ها را مشاهده کنید. جدول صحت مدار را کامل نمایید.

سؤال ۱: خروجی B₀ بیت فرضی در کدام حالت روشن می‌شود؟

- ☐ الف) $A=B$
☐ ب) $A > B$
☐ ج) $A < B$
☐ د) $A=B$ و $A < B$

با استفاده از منابع مختلف در مورد متمم ۲ عدد باینری تحقیق کرده و به کلاس ارائه نمایید.

تحقیق کنید



کار عملی ۲



بستن مدارهای جمع‌کننده و تفریق‌کننده چهار بیتی با نرم‌افزار

هدف: کسب مهارت در بررسی عملکرد مدارهای محاسبه گر (جمع و تفریق اعداد باینری)

مواد، ابزار و تجهیزات: لوازم التحریر - اینترنت - رایانه - نرم‌افزار مولتی سیم یا هر نرم‌افزار مناسب دیگر - برگه اطلاعات ICهای دیجیتال (IC ۷۴۸۳)

مراحل اجرای کار:

برای جمع و تفریق دو عدد چهار بیتی A و B می‌توانید از مدار شکل ۱۲ استفاده کنید.

نکته مهم

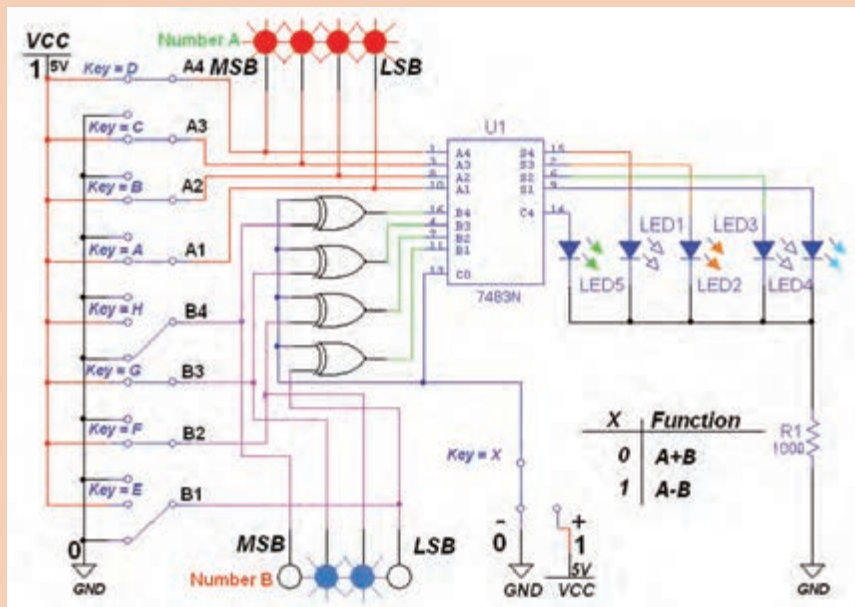


پژوهش



هنگام عمل تفریق (A-B)، خط کنترل X باید برابر با $X=1$ باشد تا متمم ۲ عدد B به دست آید و سپس عدد A جمع شود.

در رابطه با تفریق اعداد باینری به کمک متمم ۲ تحقیق کنید و نتیجه را در قالب یک گزارش به کارگاه ارائه دهید.



شکل ۱۲- مدار جمع کننده و تفریق کننده چهار بیتی

جدول ۷- جدول صحت جمع و تفریق دو عدد چهار بیتی A و B

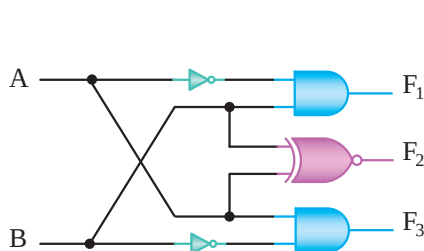
وضعیت بیت‌های عدد A					وضعیت بیت‌های عدد B					X=0					عدد حاصل	X=1					عدد حاصل
عدد A	A _f	A _r	A _r	A _l	عدد B	B _f	B _r	B _r	B _l	C _f	S _f	S _r	S _r	S _l		C _f	S _f	S _r	S _r	S _l	
۹	۱	۰	۰	۱	۶	۰	۱	۱	۰	۰	۱	۱	۱	۱	۱۵	۰	۰	۰	۱	۱	۳
	۱	۰	۰	۱		۰	۱	۱	۰												
	۱	۱	۰	۰		۰	۱	۰	۱												
	۱	۰	۰	۱		۰	۰	۱	۱												
	۰	۰	۰	۱		۱	۱	۱	۰												

۳ مدارهای مقایسه‌کننده

مقایسه‌کننده مداری است که می‌تواند دو عدد A و B باینری را با یکدیگر مقایسه نماید و دو عدد را از نظر $A < B$ ، $A = B$ و $A > B$ تشخیص دهد.

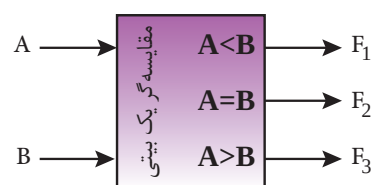
الف) مقایسه‌کننده تک بیتی: مقایسه‌کننده تک بیتی مداری است که به کمک آن می‌توان دو عدد A و B باینری یک رقمی را با یکدیگر مقایسه نمود. در صورتی که هر یک از حالات اتفاق بیفتد، خروجی مربوط به آن ۱ منطقی می‌شود، (شکل ۱۳).

جدول ۸- جدول صحت مقایسه دو بیت A و B



شکل ۱۳- مدار مقایسه‌کننده تک بیتی

A	B	F_1 $A < B$	F_2 $A = B$	F_3 $A > B$
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0



شکل ۱۴- بلوک دیاگرام مدار مقایسه‌کننده تک بیتی

با توجه به جدول ۸ تابع منطقی F_1 ، F_2 و F_3 را بنویسید.

$F_1 =$

$F_2 =$

$F_3 =$

فعالیت



با مراجعه به رسانه‌های مختلف شماره IC مقایسه‌کننده ۴ بیتی را شناسایی کنید. جدول صحت و نحوه بستن مدار آن را در قالب یک گزارش ارائه دهید.

پژوهش



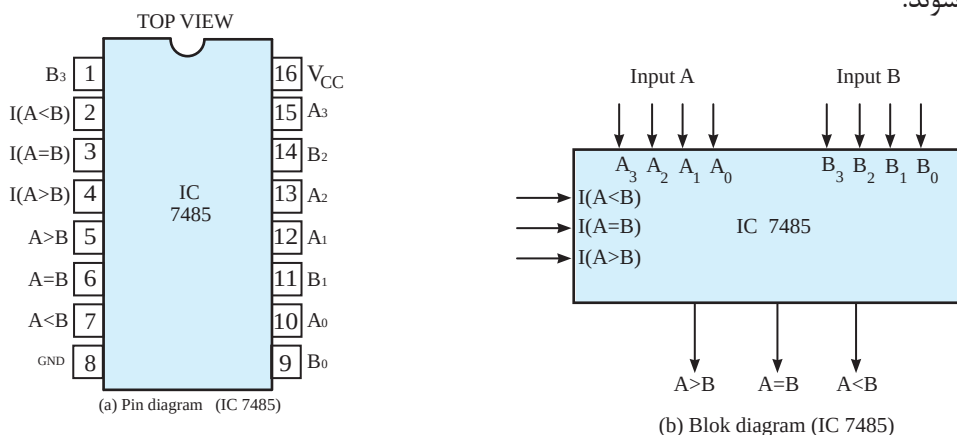
در فرایند ورودی کارت اعتباری به دستگاه ATM آیا مقایسه‌ای صورت می‌گیرد؟ توضیح دهید.

پرسش



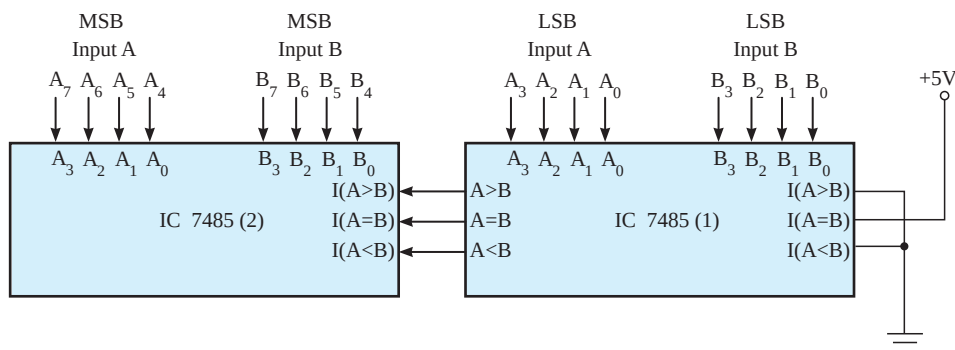
ب) مقایسه‌کننده چند بیتی: برای مقایسه دو عدد باینری چند رقمی از مدار مقایسه‌کننده چند بیتی که متشکل از چند مدار مقایسه‌کننده تک بیتی است استفاده می‌شود. برای اتصال این مدار می‌توان از گیت‌های منطقی یا IC خاص استفاده کرد.

در شکل ۱۵، IC ۷۴۸۵ که یک مقایسه‌کننده چند بیتی نشان داده شده است از این IC می‌توان برای مقایسه دو عدد باینری ۴ بیتی استفاده کرد؛ به عنوان مثال اگر بخواهیم حالت تساوی دو عدد را داشته باشیم باید ورودی‌های $I(A < B)$ و $I(A > B)$ به زمین (GND) و ورودی $I(A = B)$ به ولتاژ تغذیه (V_{CC}) متصل شوند.



شکل ۱۵

این IC ها قابلیت آبشاری شدن (Cascade) دارند و می‌توان با اتصال چند IC ۷۴۸۵ به یکدیگر، کلمات باینری با طول تقریباً دلخواه را مقایسه کرد (شکل ۱۶). ورودی‌های ۴ بیتی این IC شامل A_3 تا A_0 و B_3 تا B_0 هستند که دارای وزن باینری می‌باشند؛ به طوری که A_3 و B_3 پرارزش‌ترین بیت‌ها (MSB) محسوب می‌شوند.



شکل ۱۶

همان‌طور که گفته شد، وظیفه این IC مقایسه دو عدد ۴ بیتی و اعلام برابر بودن یا نبودن آنهاست. از این قابلیت می‌توان برای ساخت یک قفل رمزی ساده یا درب هوشمند با کارت دیجیتالی استفاده کرد.

با بهره‌گیری از منابع مختلف و کمک معلم خود مدار یک مقایسه‌کننده دو عدد دو بیتی را رسم و طرز کار آن را بررسی کنید.

فعالیت





بستن مدارهای مقایسه‌کننده با نرم‌افزار

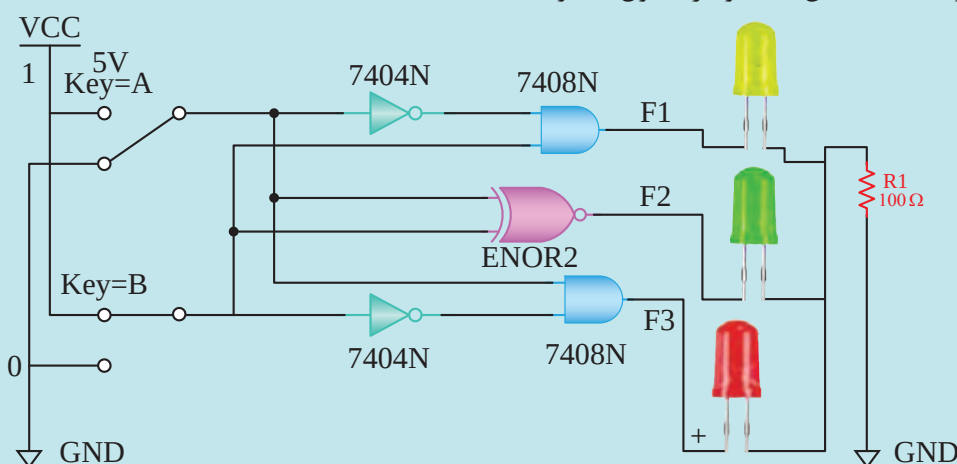
هدف: کسب مهارت بستن مدار قفل رمز (کارت دیجیتال) با مقایسه‌کننده چهار بیتی به صورت نرم‌افزاری با IC ۷۴۸۵

مواد، ابزار و تجهیزات: میز آزمایشگاهی الکترونیک با تجهیزات استاندارد - لوازم التحریر - اینترنت - رایانه - نرم‌افزار مولتی سیم یا هر نرم‌افزار مناسب دیگر - دیود نورانی LED زرد، قرمز و سبز هر کدام یک عدد - مقاومت‌های 100Ω ، $\frac{1}{4}$ وات - ترانزیستور BC۵۴۷ یک عدد - رله یک کنتاکت یک عدد، لامپ ۱۲ ولتی یک عدد - بردبرد - سیم‌های بردبرد - کلید (Dip Switch) قابل نصب روی بردبرد ۲ عدد - برگه اطلاعات IC های دیجیتال - IC ۷۴۸۵

مراحل اجرای کار:

الف) مدار مقایسه‌کننده تک بیتی

- ۱ مدار مقایسه‌گر یک بیتی شکل ۱۷ را روی محیط کار نرم‌افزار ببندید.
- ۲ ورودی A و B را طبق جدول صحت ۹، تغییر وضعیت دهید و وضعیت خروجی‌های مدار را با توجه به ورودی‌ها مشخص کنید و در جدول ۹ بنویسید.



شکل ۱۷- مدار عملی مقایسه‌کننده یک بیتی

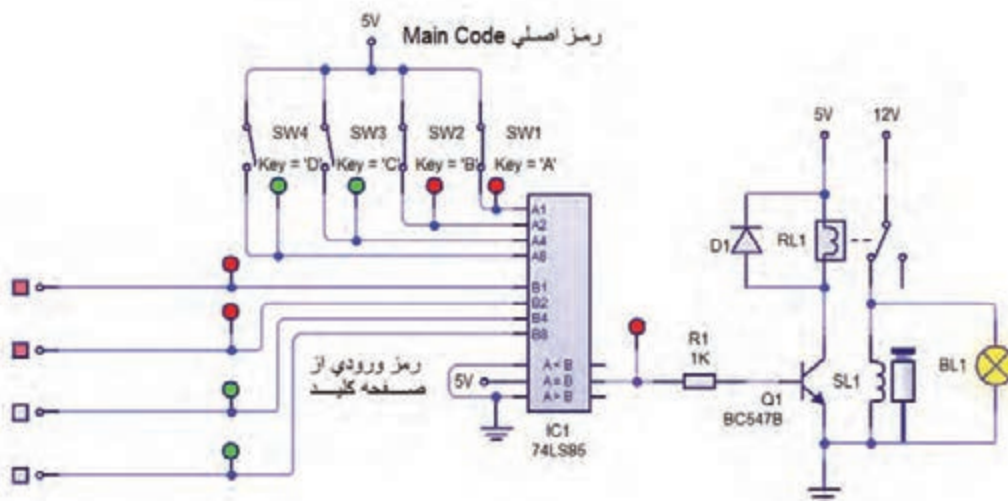
جدول ۹- جدول صحت مقایسه‌کننده تک بیتی

ورودی‌ها		F_1 $A < B$	F_2 $A = B$	F_3 $A > B$
A	B			
۰	۰			
۰	۱			
۱	۰			
۱	۱			

ب) مدار مقایسه کننده چند بیتی با IC

۱- شرح عملکرد مدار شکل ۱۸ در ورودی اول IC (A_0 تا A_3) یک عدد ثابت به عنوان رمز عبور قرار می دهیم. این عدد می تواند توسط کلیدهای (Dip Switch) یا اتصال مستقیم به V_{CC} و GND تنظیم شود. در ورودی دوم (B_0 تا B_3) عددی وارد می شود که کاربر از طریق کلید فشاری (PB) یا سوئیچ وارد می کند. پایه های $A < B$ و $A > B$ سی غیرفعال شده و فقط خروجی $A = B$ مورد استفاده قرار می گیرد. اگر عدد وارد شده با رمز ذخیره شده برابر باشد، خروجی $A = B$ فعال (۱ منطقی) می شود. این خروجی می تواند به یک LED یا رله باترانزیستور وصل شود تا در صورت درست بودن رمز، قفل باز شود (مثلاً LED روشن شود یا رله عمل کند). در غیر این صورت، مدار در حالت قفل باقی می ماند.

Input Number A	Input Number B	Output Pin 5 ($A > B$)	Output Pin 6 ($A = B$)	Output Pin 7 ($A < B$)
0000	0000	0	1	0
1000	0000	1	0	0
0000	1000	0	0	1

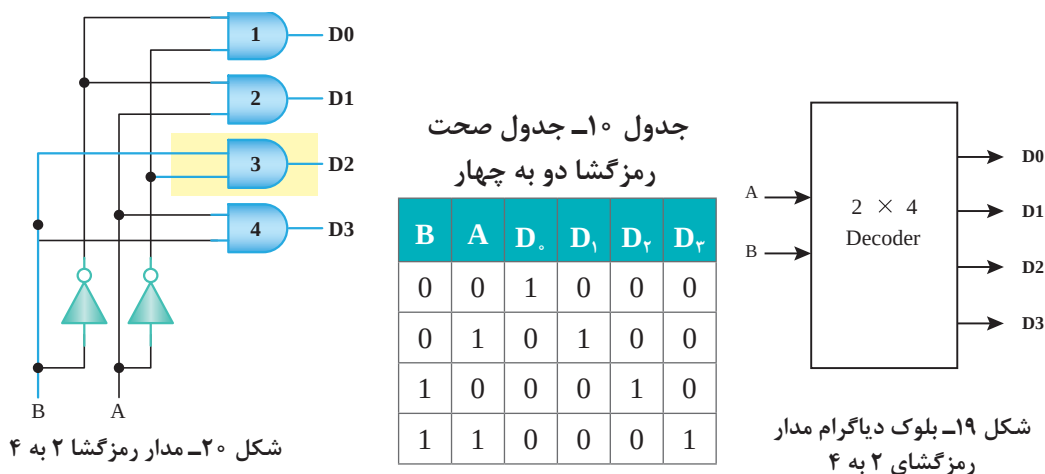


شکل ۱۸

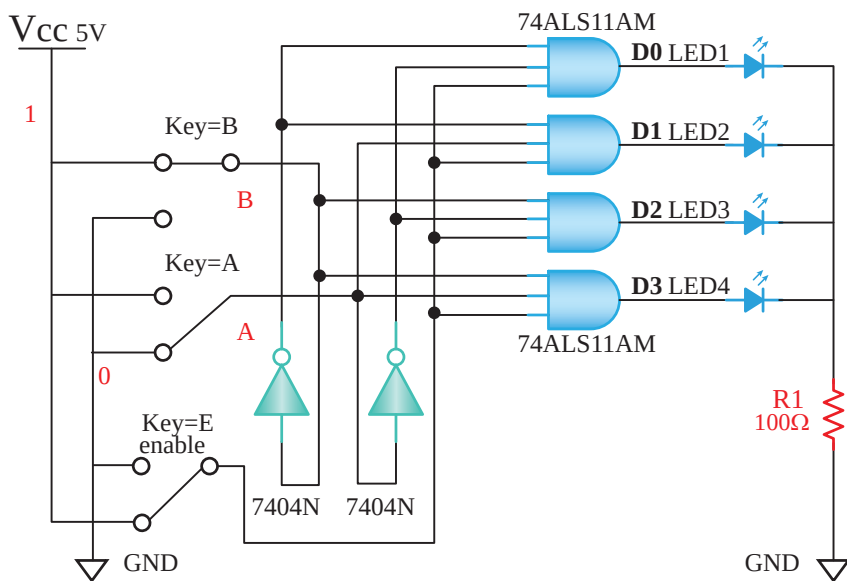
۴ مدارهای رمزگشا

برای دریافت اطلاعات از دستگاه های محاسباتی دیجیتالی مداری مورد نیاز است که اطلاعات را از حالت دودویی به اعشاری تبدیل کند. شکل ۱۹ خروجی این مدارها معمولاً به نمایشگرها متصل می شود. این تبدیل کننده ها را رمزگشا و عملی را که انجام می دهند رمزگشایی می نامیم. برای کنترل 2^n خط خروجی با n خط ورودی از مدار رمزگشا استفاده می شود.

مدار شکل ۲۰ یک رمزگشای ۲ به ۴ با خروجی در حالت "یک فعال را نشان می‌دهد.



ورودی‌های A و B را ورودی‌های آدرس و خروجی‌های D_0 ، D_1 ، D_2 و D_3 را خروجی‌های داده می‌نامیم. همان‌طور که در شکل دیده می‌شود، خروجی دروازه شماره ۳ فقط به ازای ترکیب ورودی $A=0$ و $B=1$ فعال (۱) می‌شود. برای مثال می‌توان چهار دستگاه دیجیتالی مستقل را با دو خط باینری A و B کنترل (روشن یا خاموش) کرد (شکل ۲۱).



نکته



LEDهای D_0 تا D_7 که خروجی مدار هستند به عنوان دستگاه‌های دیجیتالی در نظر گرفته شده است.

پرسش



آیا با فشار دادن دکمه‌ای مربوط به عدد در صفحه کلید تلفن و ظاهر شدن عدد مربوطه روی صفحه نمایش عمل رمزگشایی رخ می‌دهد؟ توضیح دهید.

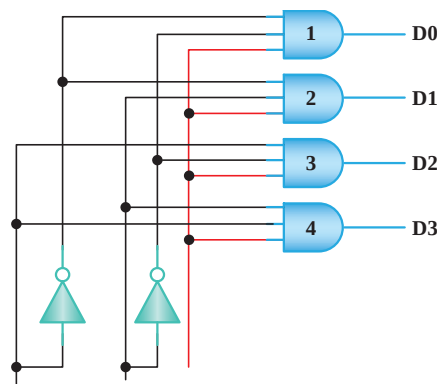
در مدارهای ترکیبی می‌توان با یک خط کنترل به نام تواناساز (Enable) خروجی را تحت کنترل درآورد. با فعال شدن پایه En خروجی‌ها می‌توانند فعال شوند. در صورتی که ورودی تواناساز En صفر باشد، خروجی فعال نخواهد شد. مدار رمزگشای دو به چهار شکل ۲۱ و ۲۲ دارای ورودی تواناساز است. که جدول صحت آن را در جدول ۱۱ مشاهده می‌کنید.

توضیح:

در جدول ۱۱ حرف X نشان‌دهنده آن است که ورودی‌ها هرچه باشد (۰ یا ۱) در آن حالت بر روی نتایج خروجی اثری ندارد.

جدول ۱۱- جدول صحت رمزگشا دو به چهار

C	B	A	D_0	D_1	D_2	D_3
0	X	X	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1



شکل ۲۲- مدار رمزگشا دو به چهار با خط ورودی تواناساز

تحقیق کنید



برای توسعه خطوط ورودی و خروجی مدارهای رمزگشا از کدام شماره ICهای TTL می‌توان استفاده کرد؟ یک نمونه با مشخصات پایه‌ها و جدول صحت آن به کارگاه ارائه دهید.

یکی دیگر از کاربردهای مهم مدارهای رمزگشا اجرای توابع منطقی است. زیرا استفاده از این مدارها سبب کاهش گیت‌های منطقی و ساده‌سازی حجم مدارهای دیجیتالی می‌شود.

فعالیت



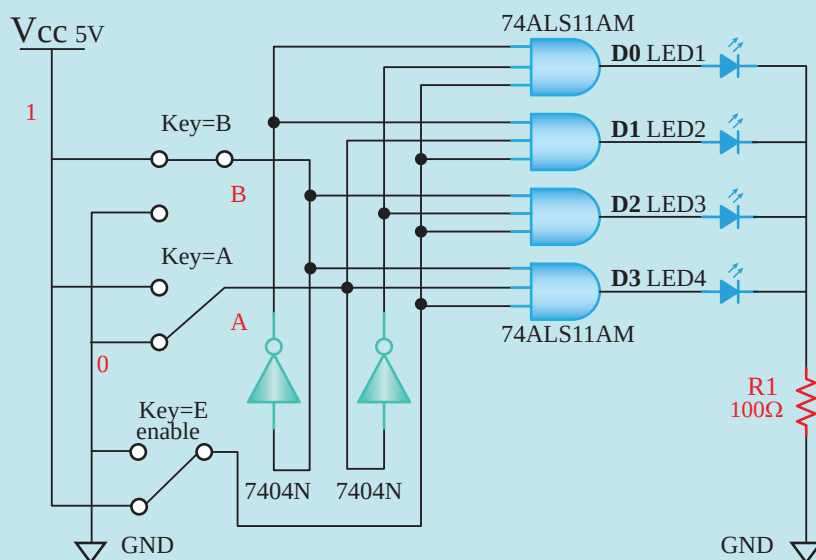
با مراجعه به کتاب همراه هنرجو و اجرای تابع با مدار رمزگشا فرایند را تحلیل کنید و مدار آن را با استفاده از نرم افزار شبیه‌ساز به اجرا درآورید.



بستن مدار رمزگشایی با نرم افزار

هدف: بررسی مدار رمزگشایی دو به چهار و تکمیل جدول درستی آن
مواد، ابزار و تجهیزات: لوازم التحریر - رایانه - نرم افزار مولتی سیم یا هر نرم افزار مناسب دیگر
مراحل اجرای کار:

- ۱ نرم افزار را فعال کنید.
- ۲ مدار شکل ۲۳ را روی محیط کار نرم افزار ببندید.
- ۳ ورودی تواناساز (En) را در صفر قرار دهید سپس ورودی A و B را تغییر وضعیت دهید آیا در خروجی‌ها تغییر حاصل می‌شود؟
- ۴ ورودی تواناساز (En) را در وضعیت (۱) صفر قرار دهید. ورودی A و B را طبق جدول ۱۲ صحت تغییر وضعیت دهید و وضعیت خروجی‌های مدار را با توجه به ورودی‌ها مشخص کنید و در جدول ۱۲ بنویسید.



شکل ۲۳- مدار عملی رمزگشای ۲ به ۴

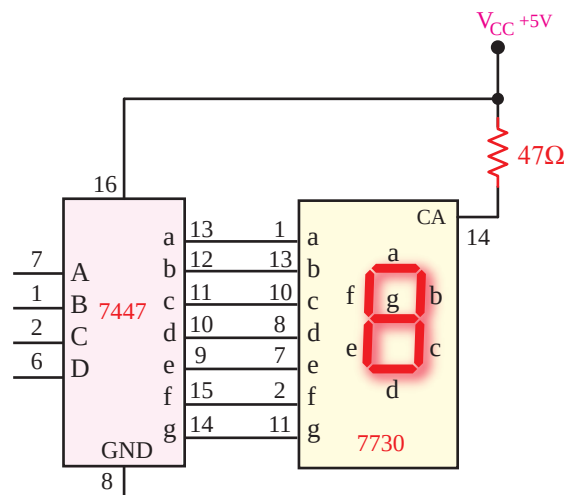
جدول ۱۲- جدول صحت رمزگشای ۲ به ۴

E_n	B	B	D_0	D_1	D_2	D_3
0	X	X				
1	0	0				
1	0	1				
1	1	0				
1	1	1				

■ مبدل BCD به 7Segment (نمایشگر هفت قطعه‌ای)

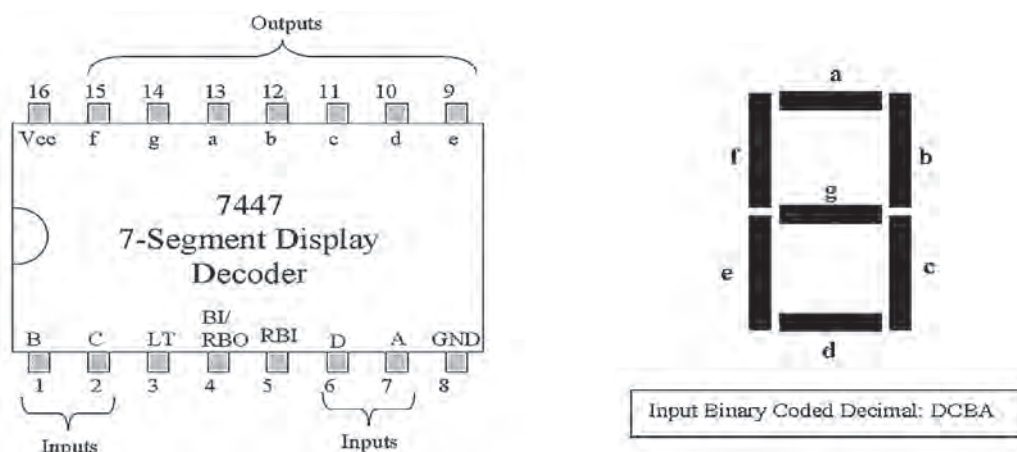
مدار منطقی مبدل BCD به 7Seg یکی از مدارهای رمزگشا است که در سیستم‌های دیجیتالی کدهای BCD را به نمایشگر هفت قطعه‌ای رمزگشایی می‌کند. برای نمایش ارقام ۰ تا ۹ و برخی حروف به کار می‌رود. برای نمایش کد BCD در 7Seg باید از IC ۷۴۴۷ (برای 7Seg آند مشترک) و IC ۷۴۴۸ (برای 7Seg کاتد مشترک) استفاده شود.

شکل ۲۴ مدار مربوط به این مبدل را نشان می‌دهد. رقم BCD را به ورودی‌های آی سی ۷۴۴۷ می‌دهیم. پایه‌های A, B, C, و D ورودی‌های IC هستند و پایه‌های a, b, c, d, e, f, و g خروجی‌های IC که به 7Seg اتصال می‌یابند. ورودی D با ارزش‌ترین رقم BCD و ورودی A کم ارزش‌ترین رقم BCD را دریافت می‌کند.



شکل ۲۴ مبدل BCD به 7Seg

IC ۷۷۳۰ نمایشگر هفت قطعه‌ای آند مشترک است. پایه ۱۴ آند مشترک (CA) و تمام LEDها است. برای محدود کردن جریان و جلوگیری از آسیب رسیدن به نمایشگر هفت قطعه‌ای 7Seg یک مقاومت 47Ω اهمی را به صورت سری با V_{CC} و در مسیر پایه مشترک ۱۴ قرار می‌دهیم.



شکل ۲۵

Inputs								Outputs						
Decimal or Function	Inputs						BI/RBO							
	LT	RBI	D	C	B	A		a	b	c	d	e	f	g
0	1	1	0	0	0	0	1	ON	ON	ON	ON	ON	ON	OFF
1	1	any	0	0	0	1	1	OFF	ON	ON	OFF	OFF	OFF	OFF
2	1	any	0	0	1	0	1	ON	ON	OFF	ON	ON	OFF	ON
3	1	any	0	0	1	1	1	ON	ON	ON	ON	OFF	OFF	ON
4	1	any	0	1	0	0	1	OFF	ON	ON	OFF	OFF	ON	ON
5	1	any	0	1	0	1	1	ON	OFF	ON	ON	OFF	ON	ON
6	1	any	0	1	1	0	1	OFF	OFF	ON	ON	ON	ON	ON
7	1	any	0	1	1	1	1	ON	ON	ON	OFF	OFF	OFF	OFF
8	1	any	1	0	0	0	1	ON	ON	ON	ON	ON	ON	ON
9	1	any	1	0	0	1	1	ON	ON	ON	OFF	OFF	ON	ON
10	1	any	1	0	1	0	1	OFF	OFF	OFF	ON	ON	OFF	ON
11	1	any	1	0	1	1	1	OFF	OFF	ON	ON	OFF	OFF	ON
12	1	any	1	1	0	0	1	OFF	ON	OFF	OFF	OFF	ON	ON
13	1	any	1	1	0	1	1	ON	OFF	OFF	ON	OFF	ON	ON
14	1	any	1	1	1	0	1	OFF	OFF	OFF	ON	ON	ON	ON
15	1	any	1	1	1	1	1	OFF	OFF	OFF	OFF	OFF	OFF	OFF
BI	any	any	any	any	any	any	0	OFF	OFF	OFF	OFF	OFF	OFF	OFF
RBI	1	0	0	0	0	0	0	OFF	OFF	OFF	OFF	OFF	OFF	OFF
LT	0	any	any	any	any	any	1	ON	ON	ON	ON	ON	ON	ON

Notes:

- BI must be high when output functions 0 through 15 are desired. RBI must be high if blanking of a decimal zero is *not* desired.
- If BI is low, all 7 segments are off, regardless of any other inputs (such as A, B, C, or D).
- The RBO is typically high. If A, B, C, D, and RBI are all low, and the lamp test (LT) is high, then all 7 segments are off. In this situation, the RBO goes low.
- If BI is high, and LT is low, all 7 segments are on. This function can be used to see if all the LED segments are working.
- Note that the BI and RBO share pin #4. It is both an input and an output.

ادامه شکل ۲۵

موارد شماره‌های ۱ الی ۵ در برگه اطلاعاتی آی سی ۷۴۴۷ را ترجمه کنید و نتیجه را در کارگاه به بحث بگذارید. (شکل ۲۵).

ترجمه



کار عملی ۵



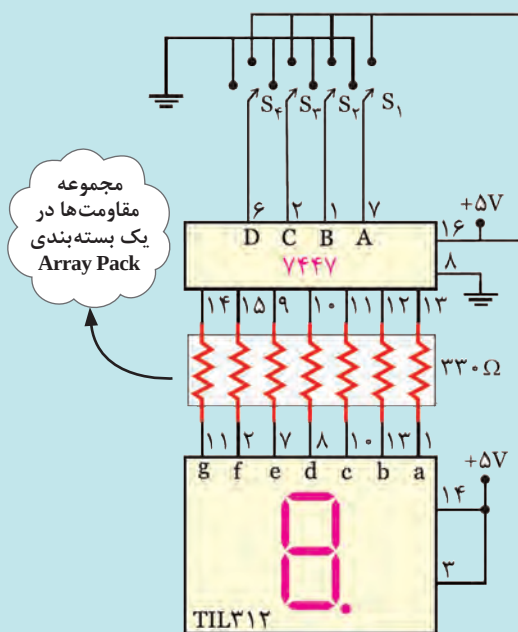
بستن مدار رمزگشا BCD به 7Seg

هدف آزمایش: بررسی عملکرد مدار رمزگشا BCD به 7Seg

مواد، ابزار و تجهیزات: منبع تغذیه ۳۰-۰ ولت ۱A یک دستگاه، 7Seg آند مشترک یک عدد، IC ۷۴۴۷ یک عدد بردبرد یک قطعه، کلیدهای SPST چهار عدد، سیم‌های رابط، مولتی‌متر دیجیتالی یک دستگاه

مراحل اجرای کار:

- اگر منبع تغذیه موجود در آزمایشگاه دارای ولتاژ ثابت ۵ ولت است از آن برای تغذیه IC استفاده کنید در غیراین صورت ابتدا منبع تغذیه را روی ۵ ولت تنظیم و سپس آن را به مدار وصل کنید.
- مدار شکل را بر روی بردبرد ببندید.



اگر کلیدهای S_1 تا S_4 در حالت ۱ قرار گیرند، ولتاژ ۵ ولت به ورودی ۷۴۴۷ IC می‌رسد. چنانچه این کلیدها در حالت صفر باشند، صفر ولت یا صفر منطقی به ورودی ۷۴۴۷ IC داده می‌شود. مثلاً برای نمایش عدد ۵ بر روی 7Seg، کلیدهای S_1 و S_3 در وضعیت یک و کلیدهای S_4 و S_2 در وضعیت صفر قرار می‌گیرند. عدد باینری ۰۱۰۱ معادل عدد دسیمال ۵ است. با قراردادن کلیدهای S_1 تا S_4 در حالت‌های مختلف جدول ۱۳ را تکمیل کنید و عدد نمایش داده شده بر روی هفت قطعه‌ای 7Seg را در جدول یادداشت نمایید.

شکل ۲۶- مدار عملی رمزگشای BCD به 7Seg

جدول ۱۳- جدول صحت مدار رمزگشای BCD به 7Seg

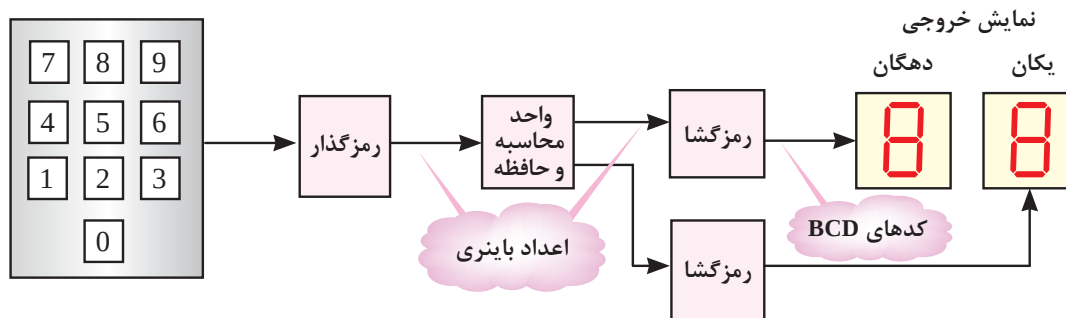
سطر	S_1	S_2	S_3	S_4	پایه‌های 7Seg a b c d e f g							نمایش عدد
					a	b	c	d	e	f	g	
0	0	0	0	0	1	1	1	1	1	1	0	
1	0	0	0	1								
2	0	0	1	0								
3	0	0	1	1								
4	0	1	0	0								
5	0	1	0	1								
6	0	1	1	0								
7	0	1	1	1								
8	1	0	0	0								
9	1	0	0	1								
10	1	0	1	0								
11	1	0	1	1								
12	1	1	0	0								
13	1	1	0	1								
14	1	1	1	0								
15	1	1	1	1								

یک رمزگشای $4 \rightarrow 2$ با دروازه‌های AND را با خط تواناساز صفر فعال رسم کنید.



۵ مدارهای رمزگذار (انکدر - Encoder):

اعدادی که به رایانه یا سامانه دیجیتال داده می‌شود، در سیستم دهدهی هستند. چون رایانه با اعداد باینری کار می‌کنند، اعداد دهدهی باید به اعداد باینری تبدیل شوند. مداری که اطلاعات را از حالت دهدهی به باینری تبدیل می‌کند، رمزگذار نام دارد. شکل ۲۷ یک سامانه دیجیتال را نشان می‌دهند.



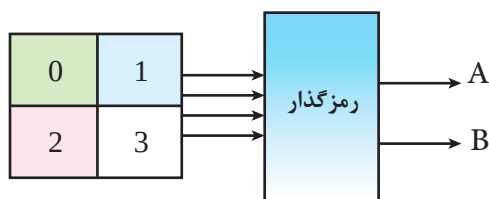
شکل ۲۷- یک سامانه دیجیتال

یک رمزگذار با $m = 2^n$ ورودی دهدهی، دارای n خروجی است.

برای مثال یک صفحه کلید با شماره‌های صفر تا سه را می‌توان به دو خط باینری تبدیل کرد. در شکل ۲۸ بلوک دیاگرام یک رمزگذار $4 \rightarrow 2$ به همراه جدول صحت در جدول ۱۴ نشان داده شده است.

جدول ۱۴- جدول صحت رمزگذار $4 \rightarrow 2$

I_3	I_2	I_1	I_0	B	A
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1

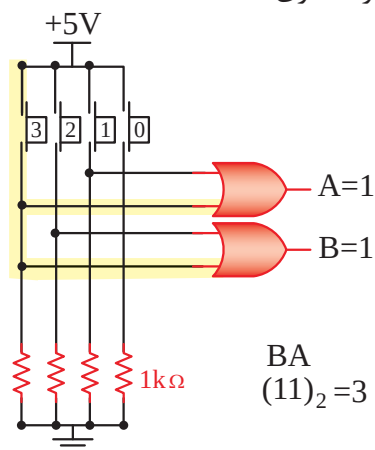


شکل ۲۸- بلوک دیاگرام رمزگذار $4 \rightarrow 2$

همانطور که مشاهده می‌کنید، در هر لحظه باید فقط یکی از ورودی‌های رمزگذار در حالت فعال قرار گیرد تا درست عمل کند.

مدار رمزگذار $4 \rightarrow 2$ را می‌توان مطابق شکل ۲۹ طراحی کرد. چنانچه کلید ۳ فشرده شود یکی از خطوط دو گیت OR «۱» منطقی می‌شود که سبب ایجاد عدد باینری سه «۱۱» در خروجی رمزگذار می‌شود.

جدول صحت این مدار را در جدول ۱۴ آورده‌ایم. توجه کنید که در این مدار برای رقم صفر، ورودی خاصی در نظر گرفته نشده است؛ زیرا زمانی که همه کلیدهای ۱ تا ۹ قطع باشند، خروجی‌های مدار در حالت «...» منطقی قرار می‌گیرند که معادل صفر اعشاری است.



شکل ۲۹- مدار رمزگذار ۴→۲

در مدار شکل ۲۹ اعداد باینری صفر تا دو چگونه ظاهر می‌شوند؟ توضیح دهید.

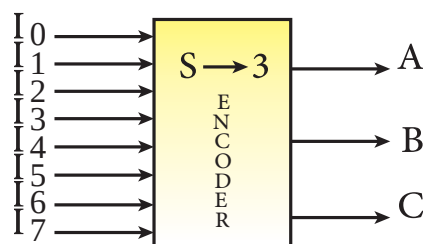
تمرین کلاسی



در شکل ۳۰ دیاگرام یک رمزگذار ۳→۸ (بخوانید ۸ به ۳) و در جدول ۱۵ جدول صحت آن نشان داده شده است. همان‌طور که مشاهده می‌کنید، در هر لحظه باید فقط یکی از ورودی‌های رمزگذار در حالت فعال باشد.

جدول ۱۵- جدول صحت یک رمزگذار ۳→۸

ورودی‌ها								خروجی‌ها		
I_v	I_e	I_d	I_f	I_r	I_t	I_l	I_o	C	B	A
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1



شکل ۳۰- بلوک دیاگرام رمزگذار ۳→۸

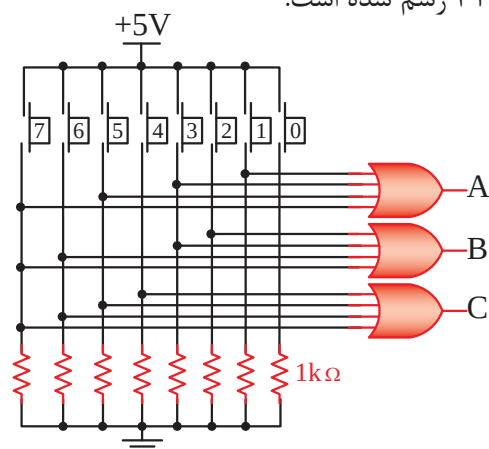
مطابق جدول ۱۵، معادلات بول هر یک از خروجی‌های A، B و C را می‌نویسیم:

$$A = I_1 + I_7 + I_8 + I_9$$

$$B = I_7 + I_8 + I_9 + I_{10}$$

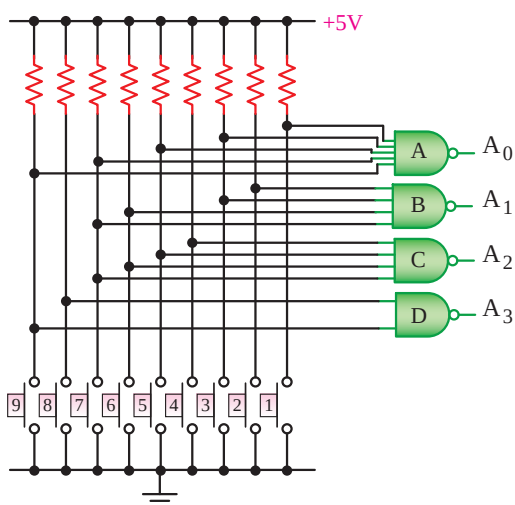
$$C = I_7 + I_8 + I_9 + I_{10}$$

مدار این رمزگذار در شکل ۳۱ رسم شده است.



شکل ۳۱- مدار رمزگذار ۸ → ۳

مدارهای رمزگذار را می‌توان با دروازه‌های منطقی NAND نیز طراحی کرد. در این صورت حالت فعال ورودی‌ها «۰» خواهد بود. در شکل ۳۲ مدار یک رمزگذار ده‌دهی به باینری که با دروازه‌های NAND اجرا شده است.



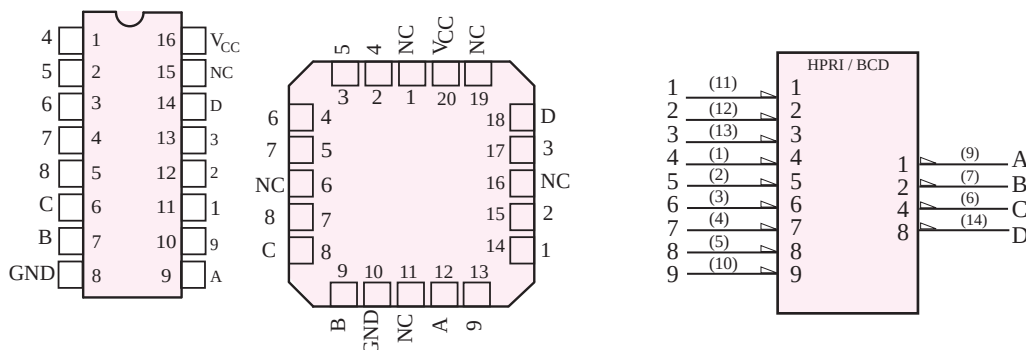
شکل ۳۲- رمزگذار ده‌دهی به باینری

جدول ۱۶- جدول صحت رمزگذار دهدهی به باینری

INPUTS									OUTPUTS			
۱	۲	۳	۴	۵	۶	۷	۸	۹	A _۳	A _۲	A _۱	A _۰
0	1	1	1	1	1	1	1	1	0	0	0	1
1	0	1	1	1	1	1	1	1	0	0	1	0
1	1	0	1	1	1	1	1	1	0	0	1	1
1	1	1	0	1	1	1	1	1	0	1	0	0
1	1	1	1	0	1	1	1	1	0	1	0	1
1	1	1	1	1	0	1	1	1	0	1	1	0
1	1	1	0	1	1	0	1	1	0	1	1	1
1	1	1	1	1	1	1	0	1	1	0	0	0
1	1	1	1	1	1	1	1	0	1	0	0	1
1	1	1	1	1	1	1	1	1	0	0	0	0

همان گونه که متذکر شدیم، در یک لحظه معین فقط یکی از ورودی‌های رمزگذار می‌تواند فعال شود؛ مثلاً در شکل ۳۲ اگر کلید ۹ فشرده شود، در خروجی مدار ترکیب باینری هم‌ارز آن یعنی ۱۰۰۱ ظاهر می‌شود. حال اگر به‌طور اتفاقی کلیدهای ۶ و ۹ هم‌زمان فشرده شوند، در خروجی مدار ترکیب باینری ۱۱۱۱ ظاهر می‌شود که معادل هیچ کدام از ارقام ۶ یا ۹ نیست. برای پیشگیری از بروز این گونه خطاها، رمزگذارهای صنعتی را طوری طراحی می‌کنند که در صورت فشرده شدن هم‌زمان چند کلید، فقط ترکیب باینری هم‌ارز بزرگ‌ترین رقم اعشاری را تولید کنند. این رمزگذارها را رمزگذارهای با رعایت تقدم (Priority Encoder) می‌نامند.

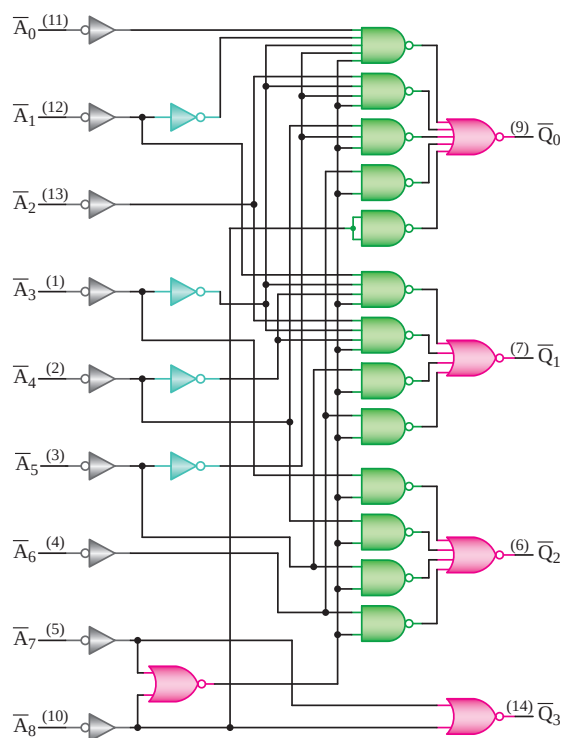
مثلاً اگر دو کلید ۶ و ۹ هم‌زمان فشرده شوند، رمزگذار صنعتی ارزش کلید ۹ را تولید می‌کند. تراشه SN۷۴۱۴۷ (CI) یک رمزگذار دهدهی به باینری با رعایت تقدم است. مشخصات کامل آن را در شکل ۳۳ مشاهده می‌کنید. در شکل ۳۳- الف علامت استاندارد IC، در شکل ۳۳- ب نمای بالا و شماره پایه‌های IC و در شکل ۳۳- پ مدارهای داخلی آن را ملاحظه کنید.



(ب) نمای بالا و شماره پایه‌ها

الف) علامت استاندارد IEEE/ANSI

شکل ۳۳- رمزگذار تقدمی SN۷۴۱۴۷



پ) مدارهای داخلی

ادامه شکل ۳۳

جدول ۱۷ جدول صحت IC شماره SN۷۴۱۴۷ را نشان می‌دهد.

جدول ۱۷- جدول صحت رمزگذار تقدمی SN۷۴۱۴۷

INPUTS									OUTPUTS			
۱	۲	۳	۴	۵	۶	۷	۸	۹	A _r	A _r	A _۱	A _۰
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	L	H	H
X	X	L	H	H	H	H	H	H	H	H	L	L
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L



بستن مدار رمزگذار با نرم افزار

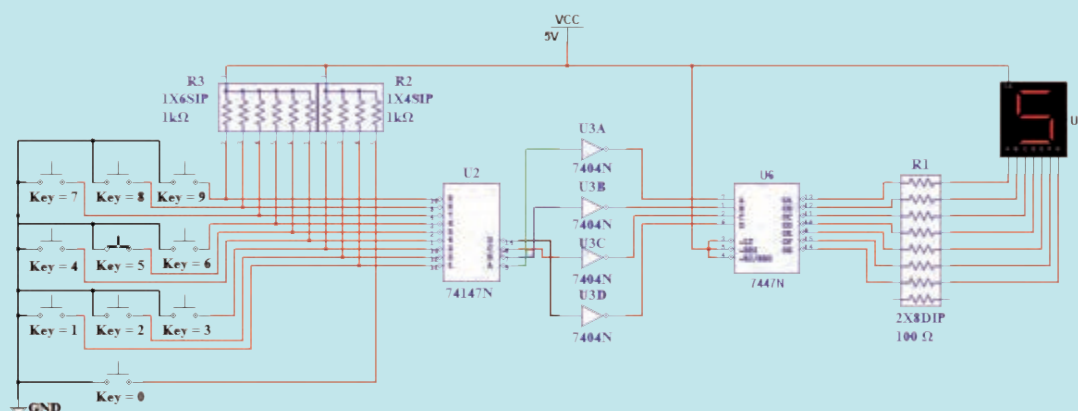
هدف: بررسی مدار رمزگذار با اولویت (Priority) دو به چهار و تکمیل جدول درستی آن
مواد، ابزار و تجهیزات: لوازم التحریر - رایانه - نرم افزار مولتی سیم یا هر نرم افزار مناسب دیگر
مراحل اجرای کار:

۱ شرح عملکرد مدار شکل (۳۴).

در انکودرهای بدون اولویت، اگر چند ورودی همزمان فعال شوند، خروجی مشخص و قابل پیش بینی نیست. در انکودر دارای اولویت، اگر چند ورودی به طور همزمان فعال شوند، فقط ورودی با عدد بزرگ تر در خروجی ظاهر می شود.

برای مثال، اگر همزمان کلیدهای ۳، ۵ و ۸ فشار داده شوند، تنها عدد ۸ در خروجی نمایش داده می شود. مدار شکل ۳۴ نشان می دهد که چگونه می توان از یک انکودر و دیکودر برای راه اندازی یک نمایشگر LED با صفحه کلید ۰ تا ۹ استفاده کرد. IC 74SN ۱۷۴ ورودی های صفحه کلید را به کد BCD با منطق منفی تبدیل می کند. سپس با استفاده از گیت های NOT، این کد به BCD با منطق مثبت تبدیل می شود. در ادامه این کد به IC ۷۴۴۷ داده می شود تا عدد روی نمایشگر هفت قطعه ای نشان داده شود.

۲ نرم افزار را فعال کنید.



شکل ۳۴

۳ مدار شکل ۳۴ را روی محیط کار نرم افزار ببندید.

ابتدا نقشه مدار را به دقت بررسی کرده و قطعات مورد نیاز مدار شامل IC های ۷۴۴۷، ۷۴۱۴۷N، نمایشگر هفت قطعه ای، کلیدها (یا صفحه کلید ۰ تا ۹)، مقاومت ها، گیت های NOT و منبع تغذیه ۵ ولت را به میز کار نرم افزار بیاورید. سپس اتصالات را با دقت برقرار کنید.

۴ با فشردن هر کلید از صفحه کلید، عدد مربوطه باید به درستی روی نمایشگر ظاهر شود. در صورت فشردن همزمان چند کلید، فقط عدد بزرگ تر نمایش داده می شود.

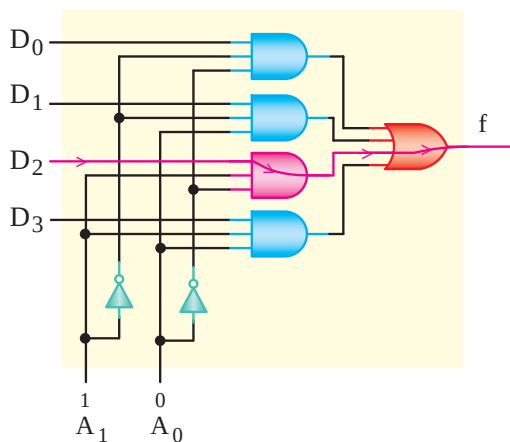
۵ در صورت نمایش نادرست اعداد، اتصالات، تغذیه و شماره آیی ها را بررسی کرده و اشکالات احتمالی را برطرف کنید.

۶ جدول ۱۸ را کامل کنید و برای ارزشیابی به هنرآموز کارگاه ارائه دهید.

جدول ۱۸

ردیف	فشردن شاسی‌های ورودی	عدد مشاهده شده در نمایشگر خروجی
۱	فقط شماره سه (۳)	
۲	فقط شماره صفر (۰)	
۳	فقط شماره هشت (۸)	
۴	فشردن هم‌زمان شاسی‌های سه و چهار	
۵	فشردن هم‌زمان شاسی‌های هشت و شش	
۶	فشردن هم‌زمان شاسی‌های پنج، هفت و نه	

۶ مدارهای مالتی پلکسر



شکل ۳۵- مالتی پلکسر ۴ → ۱

مالتی پلکسرهای دارای $m = 2^n$ خط ورودی هستند که از طریق n خط آدرس‌دهی، اطلاعات خطوط ورودی را با یک خط خروجی اتصال می‌دهند.

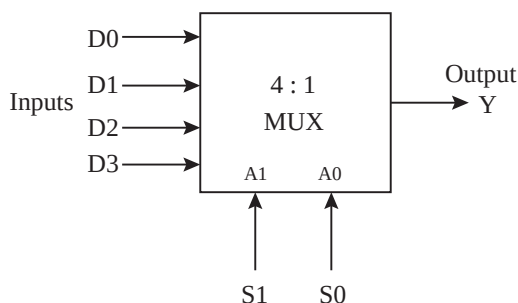
برای مثال اگر ۴ خط ورودی داشته باشیم تعداد خطوط آدرس‌دهی دو خط خواهد شد، زیرا

$m = 2^n \Rightarrow 4 = 2^2 \Rightarrow n = 2$ یعنی برای آدرس‌دهی دو خط A_1 و A_0 (یا S_1 و S_0) را در نظر می‌گیریم.

شکل ۳۵ یک مالتی پلکسر ۴ به ۱ است. مثلاً اگر کد خط آدرس $(A_1 A_0 = 10)$ باشد در این صورت ورودی $D2$ و اگر کد خط آدرس $(A_1 A_0 = 01)$ باشد، مطابق

شکل ۳۶ ورودی $D1$ را به خروجی وصل می‌کند.

Control Inputs		Output
A_1	A_0	Y
0	0	$D0$
0	1	$D1$
1	0	$D2$
1	1	$D3$

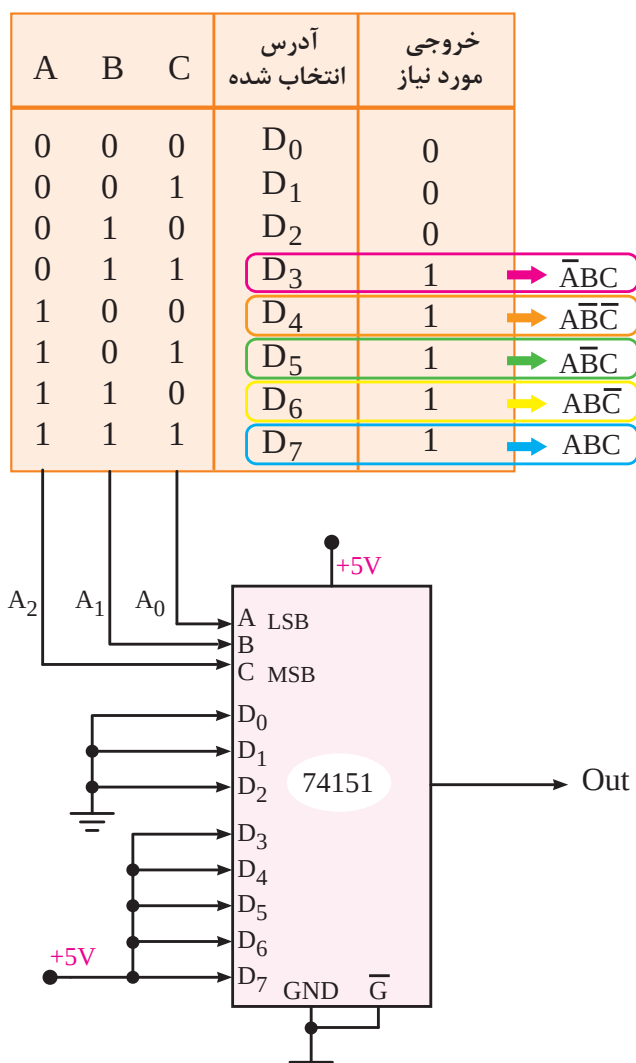


شکل ۳۶

مالتی پلکسرها با ورودی‌های بیشتر به صورت IC به بازار عرضه می‌شود. IC ۷۴۱۵۱ یک مالتی پلکسر ۸ به ۱ با سه خط آدرس‌دهی است. از جمله زمینه‌های کاربردهای مالتی پلکسر می‌توان به اجرای توابع منطقی اشاره کرد (شکل ۳۶).

در شکل ۳۷ تابع $F = \overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$ با یک IC مالتی پلکسر ۷۴۱۵۱ اجرا شده است.

چون در تابع موردنظر خروجی به ازای یک شدن جملات مینترم (۳ و ۴ و ۵ و ۶ و ۷) وجود دارد لذا ورودی‌های متناظر این جملات مینترم به یک منطقی (+۵) وصل شده‌اند و مابقی جملات تابع به صفر منطقی (زمین) متصل شده‌اند.



شکل ۳۷- مدار و جدول صحت مالتی پلکسر ۱ → ۸

پژوهش



نکته مهم

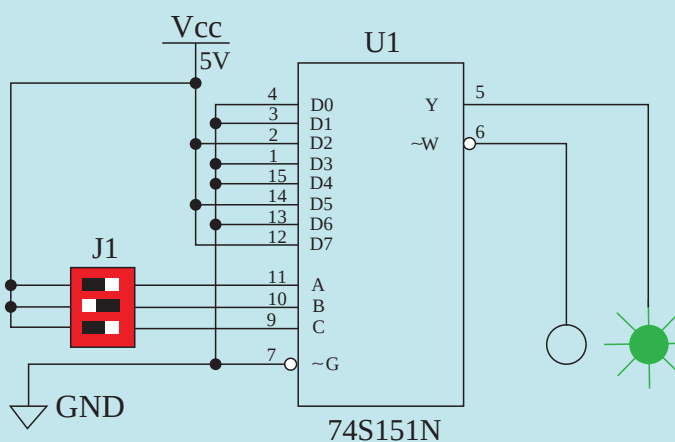


کار عملی ۷



مراحل اجرای کار:

۴ تابعی که با مالتی پلکسر اجرا شده است را با توجه به عملکرد مدار بنویسید.

[illegible]

شکل ۳۸- مدار مالتی پلکسر

۱ برخی از مدارهای ترکیبی بیش از یک خروجی دارند.

درست ☐ نادرست ☐

۲ جمع گر تک بیتی از گیت و گیت ساخته می شود.

۳ با توجه به نقشه شکل ۱۳ در مدار مقایسه کننده از کدام گیت استفاده شده است؟

الف) OR ب) XNOR پ) NAND ت) XOR

۴ از کدام مدار برای آدرس دهی حافظه ها استفاده می شود؟

الف) مالتی پلکسر ب) مقایسه کننده پ) رمزگشا ت) رمزگذار

۵ کاربرد مالتی پلکسر را به طور خلاصه شرح دهید.

۶ کدام شماره IC رمزگذار است؟

۱- ۷۴۱۴۷ ۲- ۷۴۴۷ ۳- ۷۴۱۵۱ ۴- ۷۴۸۳

۷ برای کنترل 2^n خط خروجی، به چند خط ورودی نیاز داریم؟

الف) 2^n خط ب) n خط پ) $2n$ خط ت) n^2 خط

۸ خروجی مدارهای رمزگشا معمولاً به نمایشگرها متصل می شوند.

صحیح ☐ غلط ☐

۹ برای نمایش ارقام در یک نمایشگر هفت قطعه ای از نوع آند مشترک، استفاده از کدام آی سی الزامی است؟

الف) ۷۴۴۸ ب) ۷۴۴۷ پ) مبدل دودویی به اعشاری ساده ت) رمزگشای ۲ به ۴

۱۰ تراشه IC ۷۴۱۴۷ به عنوان یک رمزگذار اولویت دار (Priority Encoder) چه ویژگی مهمی دارد؟

الف) در هر لحظه فقط یک ورودی می تواند فعال باشد.

ب) اگر چند ورودی همزمان فعال شوند، خروجی نشان دهنده ورودی با بالاترین شماره (اولویت بالاتر) است.

پ) خروجی آن مستقیماً برای راه اندازی (V-Segment) استفاده می شود.

ت) این آی سی برای تبدیل که BCD به دودویی طراحی نشده است.

ارزشیابی شایستگی پیاده‌سازی مدارهای ترکیبی

نام و نام خانوادگی		شماره ملی			نوبت:
کد حرفه	۸۲۱۲۱۲	حرفه :	مونتاز کاران تجهیزات و وسایل برقی و الکترونیکی	کد پیمان/پودمان	۸۲۱۲۱۲۰۳۱
کد وظیفه شغل	۸۲۱۲۱۲۰۳	وظیفه شغل:	ساخت مدارهای دیجیتال و میکروکنترلر	سطح صلاحیت	۲
کد کار	۸۲۱۲۱۲۰۳۳۱	کار	پیاده‌سازی مدارهای ترکیبی	سطح شایستگی	مهارت

ردیف	مراحل کار	شرایط انجام کار (ابزار، مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	نمره کسب شده
۱	طراحی مدارهای منطقی ترکیبی پایه شامل جمع‌گر، مقایسه‌گر، مالتی پلکسر، رمزگشا و رمزگذار براساس مشخصات فنی	رایانه و نرم‌افزار شبیه‌سازی مدار، مشخصات فنی، پرینتر، آی‌سی گیت منطقی، سیم رابط، منبع تغذیه، قلم و زمان ۳۰ تا ۴۰ دقیقه در کارگاه الکترونیک	بررسی مدار و جدول درستی، مشاهده عملکرد، چک‌لیست طراحی و پرسش شفاهی	طراحی و پیاده‌سازی صحیح	۳	
					۲	
					۱	
۲	تحلیل عملکرد مدارهای طراحی شده با استفاده از نرم‌افزارهای شبیه‌سازی مدار و ارزیابی نتایج	رایانه و نرم‌افزار شبیه‌سازی مدار، فایل مدار، دفترچه، کتاب، زمان ۳۰ تا ۴۰ دقیقه در سایت رایانه	شبیه‌سازی صحیح، تحلیل درست ورودی/خروجی، تفسیر نتایج و تطابق با طراحی و مشخصات فنی	شبیه‌سازی صحیح با تحلیل قابل قبول و خطای جزئی	۳	
					۲	
					۱	
۳	مونتاژ مدارهای منطقی ترکیبی با استفاده از قطعات الکترونیکی استاندارد بر روی بردبورد یا PCB	بردبرد یا PCB، آی‌سی گیت، سیم رابط، مقاومت و LED، منبع تغذیه، ابزار اندازه‌گیری و مونتاژ، نقشه مدار و زمان تقریبی ۳۰ تا ۴۰ دقیقه در کارگاه الکترونیک	مشاهده مونتاژ، تست عملکرد مدار با منبع تغذیه و ابزار اندازه‌گیری و چک‌لیست	مونتاژ قابل قبول با نقص جزئی	۳	
					۲	
					۱	
۴	انجام یک پروژه عملی شامل طراحی، شبیه‌سازی، ساخت و تست مدار منطقی ترکیبی، مستندسازی و گزارش دهی فنی	رایانه، نرم‌افزار شبیه‌سازی مدار، پرینتر یا PCB، آی‌سی مدار منطقی، سیم رابط، منبع تغذیه، ابزار مونتاژ و زمان ۴۵ تا ۶۰ دقیقه در کارگاه الکترونیک	بررسی عملکرد پروژه، مدارو شبیه‌سازی، گزارش فنی و چک‌لیست و ارائه شفاهی	اجرا، عملکرد و مستندسازی دقیق	۳	
					۲	
					۱	
	بایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:	آزمایشگاه مدارهای منطقی یا سایت رایانه، نرم‌افزار شبیه‌سازی مدار، تجهیزات کارگاهی و حفاظت فردی و دستورالعمل‌های ایمنی و بهداشت حرفه‌ای	مشاهده رفتار هنرجو در حین انجام فعالیت، میزان رعایت اصول ایمنی و نظم، فعالیت‌های گروهی، مدیریت زمان و مسئولیت‌پذیری	رعایت کامل ایمنی، بهداشت و محیط‌زیست، رفتار حرفه‌ای، نظم و مسئولیت‌پذیری بالا	۳	
					۱	
						☐ بلی
						☐ خیر
ارزشیابی کار (شایستگی انجام کار)						

توضیحات:

۱- معیار شایستگی انجام کار:

- کسب حداقل نمره ۲ در مراحل بحرانی شامل طراحی مدار منطقی ترکیبی براساس مشخصات فنی، تحلیل و شبیه‌سازی مدار، مونتاز و ساخت مدار و اجرای پروژه عملی همراه با تست و مستندسازی.

- کسب حداقل نمره ۲ از بخش شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش

- کسب حداقل میانگین ۲ از مراحل کار

۲- تعریف سطوح شایستگی: صرف‌نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

واحد یادگیری ۲: پیاده‌سازی مدارهای ترتیبی

آیا تا به حال فکر کرده‌اید

- طراحی مدارهای منطقی پیشرفته براساس یک اصول واحد صورت می‌گیرد.
 - در سامانه‌های دیجیتالی، اطلاعات طی چه مراحل و به چه صورت ثبت و ذخیره می‌شوند؟
 - در مدارهای دیجیتالی، اطلاعات چگونه در قسمت‌های مختلف جابه‌جا می‌شوند؟
 - نمایش زمان در مدارهای چراغ‌های راهنمایی دیجیتالی با چه نوع مداری صورت می‌گیرد؟
 - دستگاه فرکانس‌متر دیجیتالی چه اجزایی دارد و چگونه کار می‌کند؟
- مدارهای ترتیبی، مدارهایی منطقی هستند که خروجی آنها تابع وضعیت زمان حال و گذشته متغیرهای ورودی مدار است. در این مدارها از عناصر حافظه (فلیپ‌فلاپ) استفاده می‌شود. مدارهایی مانند حافظه‌ها، ثبات‌ها (رجیسترها)، شمارنده‌ها و ساعت دیجیتالی، مدارهای ترتیبی هستند. یادآوری می‌شود که برای تحلیل، بررسی و اجرای عملی یک مدار کاربردی، باید در تمام مراحل، نکات ایمنی و بهداشتی و توجه به مهارت‌های غیرفنی مانند کارگروهی، مسئولیت‌پذیری، پایبندی به نظم و ترتیب که از اهمیت ویژه‌ای برخوردار است، رعایت شود.

استاندارد عملکرد

پیاده‌سازی مدارهای ترتیبی شامل انواع فلیپ فلاپ (S, JK, D, T) ثبات‌ها، شمارنده‌ها و انواع آن با استفاده از محیط نرم‌افزار و بردبرد و ساخت پروژه مدار دیجیتالی.

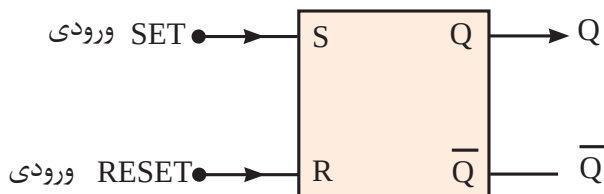
همان‌طور که مشاهده کردید خروجی مدارهای منطقی ترکیبی، در هر لحظه تابع معینی از ورودی‌های آنها در همان لحظه است. به عبارت دیگر با تغییر هم زمان ورودی‌های مدار، خروجی‌های آن نیز متناسب با ورودی‌ها تغییر می‌کند. مدارهای ترکیبی توانایی نگهداری و به‌خاطر سپاری حالت‌های ورودی را ندارند. ماشین‌های حسابگر و سیستم‌های کنترل دیجیتالی، برای نگهداری اطلاعات و استفاده مکرر از آنها نیاز به حافظه دارند. مدارهای دارای حافظه را مدارهای ترتیبی می‌نامند. این مدارها قابلیت به‌خاطر سپاری (در حافظه نگهداشتن) ترتیب پیامدها را دارند. ممکن است یک مدار ترتیبی بیش از یک ورودی یا خروجی داشته باشد. در اغلب مدارهای ترتیبی زمان تغییر وضعیت یا پذیرش اطلاعات جدید را به کمک یک سیگنال کنترل خاص، که پالس ساعت (Clock Pulse) نامیده می‌شود، تعیین می‌کنند. با چند فلیپ‌فلاپ می‌توانیم یک ثبات اطلاعات یا یک شمارنده ساخت. علاوه بر این، برای تقسیم فرکانس، ساعت و نظایر آن باید از فلیپ‌فلاپ‌ها استفاده کرد.

فلیپ‌فلاپ‌ها (Flip-Flop)

برای ذخیره اطلاعات دودویی در مدارهای دیجیتالی از حافظه‌ها استفاده می‌کنند. مدارهای ترتیبی از عناصر اصلی حافظه‌ها به‌شمار می‌روند. سلول‌های حافظه ساده‌ترین مدارهای ترتیبی هستند. به مدارهای ترتیبی که دو وضعیت پایدار دارند «فلیپ‌فلاپ» می‌گویند. فلیپ‌فلاپ‌ها در چهار دسته SR, JK, D و T وجود دارند.

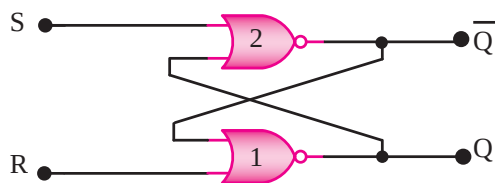
■ فلیپ‌فلاپ SR (Set Reset Flip-Flop):

یکی از این نوع مدارها، فلیپ‌فلاپ SR است. فلیپ‌فلاپ SR مدارهای ترتیبی هستند که دو وضعیت Set و Reset پایدار دارند. بلوک دیاگرام مدار فلیپ‌فلاپ SR را در شکل ۳۹ مشاهده می‌کنید.



شکل ۳۹- بلوک دیاگرام فلیپ‌فلاپ SR

مدارهای فلیپ‌فلاپ را می‌توان با استفاده از دو گیت NOR یا NAND ساخت. در اینجا برای بررسی عملکرد مدار فلیپ‌فلاپ SR فقط یک نوع آن مورد تحلیل قرار می‌گیرد. در شکل ۴۰ مدار فلیپ‌فلاپ SR با استفاده از گیت NOR را مشاهده می‌کنید.



شکل ۴۰- مدار SR-FF با گیت‌های NOR

S اول کلمه Set به معنی تنظیم کردن، پرکردن و R اول کلمه Reset به معنی بازگشت به حالت اولیه یا پاک کردن است. در مدار فلیپ‌فلاپ از خروجی یک گیت به ورودی گیت دیگر اتصال داده شده است که اصطلاحاً به این اتصال «فیدبک» یا «پس‌خورد» گفته می‌شود. این فلیپ‌فلاپ دارای دو خروجی است که یکی Q و دیگری نفی آن $\bar{Q}$ نام دارد و دو ورودی آن S و R نام دارند (شکل ۴۰).

هرچند این مدار دارای ۲ ورودی است و می‌بایست ۴ حالت را برای خروجی آن در نظر گرفت اما چون لازم است تا قبل از اعمال ورودی، شرایط اولیه هریک از خروجی‌ها که به ورودی طرف مقابل فیدبک شده را نیز در نظر بگیریم. به همین دلیل در جدول صحت با در نظر گرفتن دو حالت مختلف برای خروجی Q و $\bar{Q}$ جمعاً ۸ حالت را مشابه جدول ۲۰ بخش اول باید در نظر بگیریم.

Q_t وضعیت حافظه در لحظه کنونی (حال)

Q_{t-1} وضعیت حافظه در لحظه قبل

$Q = 1$ به مفهوم حافظه پُر است.

$Q = 0$ به مفهوم حافظه خالی است.

نکته



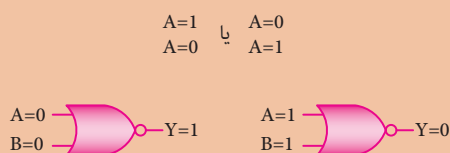
جدول ۲۰- جدول صحت فلیپ‌فلاپ SR

وضعیت ورودی‌ها		وضعیت قبلی FF-RS (خروجی اول)		وضعیت فعلی FF-RS (خروجی دوم)		
S	R	$Q_{(t-1)}$	$\bar{Q}_{(t-1)}$	$Q_{(t)}$	$\bar{Q}_{(t)}$	
0	0	0	1	0	1	فلیپ‌فلاپ وضعیت قبلی خود را حفظ می‌کند، حالت ذخیره.
0	0	1	0	1	0	
0	1	0	1	0	1	فلیپ‌فلاپ بدون توجه به حالت قبلی Reset می‌شود.
0	1	1	0	0	1	
1	0	0	1	1	0	فلیپ‌فلاپ بدون توجه به حالت قبلی Set می‌شود.
1	0	1	0	1	0	
1	1	0	1	0	0	فلیپ‌فلاپ وارد حالت غیرمجاز می‌شود.
1	1	1	0	0	0	

همان‌طوری که در شکل ۴۰ مشاهده می‌کنید خروجی گیت شماره ۱ به ورودی گیت شماره ۲ و خروجی گیت شماره ۲ به ورودی گیت شماره ۱ فیدبک شده است.

■ بررسی عملکرد مدار فلیپ‌فلاپ S-R

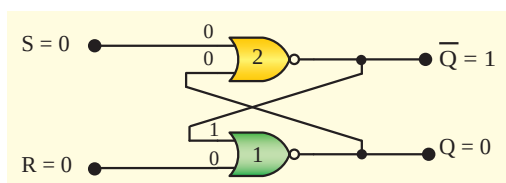
یادآوری



شکل ۴۱- گیت NOR

■ حالت اول: اگر ورودی‌های S و R هر دو در حالت

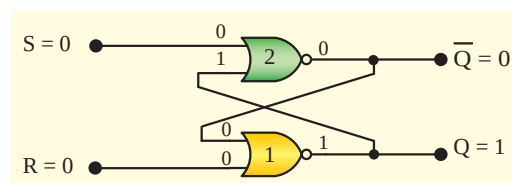
صفر منطقی باشد و حالت قبلی فلیپ‌فلاپ به صورت $Q=0$ و $\bar{Q}=1$ باشد؛ هر دو ورودی دروازه ۲ در حالت صفر قرار می‌گیرد لذا خروجی آن در حالت یک باقی می‌ماند (سطر اول جدول صحت NOR) از سوی دیگر یکی از ورودی‌های دروازه ۱ در حالت یک قرار دارد لذا خروجی این دروازه نیز در حالت صفر باقی می‌ماند. به عبارت دیگر، وضعیت خروجی‌ها تغییر نمی‌کند. شکل ۴۲ این حالت را نشان می‌دهد.

 $R=0$ و $S=0$

شکل ۴۲

■ حالت دوم: حال اگر $S=R=0$ و حالت قبلی فلیپ‌فلاپ

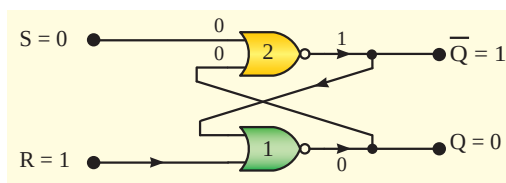
به صورت $Q=1$ و $\bar{Q}=0$ باشد، ورودی‌های دروازه ۱ هر دو در حالت صفر منطقی قرار دارد و خروجی آن در حالت یک باقی می‌ماند، از طرفی چون یکی از ورودی‌های دروازه ۲ در حالت یک منطقی است، خروجی این دروازه نیز در حالت صفر باقی می‌ماند. به این ترتیب باز هم وضعیت خروجی‌ها تغییر نمی‌کند. شکل ۴۳ این حالت را نشان می‌دهد.

 $R=0$ و $S=0$

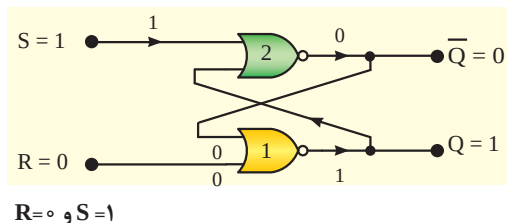
شکل ۴۳

■ حالت سوم: اکنون حالتی را در نظر می‌گیریم که

$S=0$ و $R=1$ باشد. در این حالت بدون توجه به وضعیت قبلی فلیپ‌فلاپ، به دلیل اینکه یکی از ورودی‌های دروازه ۱ در حالت یک منطقی است، خروجی این دروازه صفر می‌شود. با صفر شدن خروجی دروازه ۱ هر دو ورودی دروازه ۲ در حالت صفر منطقی قرار می‌گیرد و خروجی آن به حالت یک منطقی می‌رود. به بیان دیگر اگر $S=0$ و $R=1$ باشد، بدون توجه به وضعیت قبلی به حالت $Q=0$ و $\bar{Q}=1$ می‌رود، شکل ۴۴ این حالت را نشان می‌دهد.

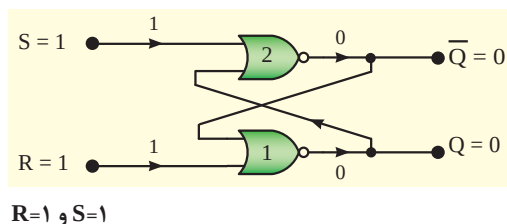
 $R=1$ و $S=0$

شکل ۴۴



شکل ۴۵

■ **حالت چهارم:** اگر $S=1$ و $R=0$ باشد، چون یکی از ورودی‌های دروازه ۲ در حالت یک منطقی است و خروجی آن به حالت صفر منطقی می‌رود. در این صورت، هر دو ورودی دروازه ۱ در حالت صفر منطقی قرار می‌گیرد و خروجی این دروازه نیز یک می‌شود به بیانی دیگر اگر $S=1$ و $R=0$ باشد، بدون توجه به وضعیت قبلی سیستم، $Q=1$ و $\bar{Q}=0$ می‌شود. شکل ۴۵ این حالت را نشان می‌دهد.



شکل ۴۶

■ **حالت پنجم:** حالتی را در نظر می‌گیریم که $S=R=1$ شود. در این حالت نیز چون دست کم یکی از ورودی‌های هر دو دروازه ۱ و ۲ در حالت یک منطقی است، خروجی‌های هر دو دروازه در حالت صفر منطقی قرار می‌گیرد. شکل ۴۶ این حالت را نشان می‌دهد. در این حالت وضع حافظه (Q) و نفی آن ($\bar{Q}$) با هم برابر شده‌اند ($Q=\bar{Q}=0$) این وضعیت را غیرمجاز می‌نامند و در جدول درستی با * نشان داده شده است.

جدول ۲۱- خلاصه جدول صحت فلیپ‌فلاپ S-R

S	R	$Q_{(t)}$
0	0	$Q_{(t-1)}$
0	1	0
1	0	1
1	1	*

اگر وضعیت خروجی‌های فلیپ‌فلاپ را قبل از تغییر حالت با $Q_{(t-1)}$ و $\bar{Q}_{(t-1)}$ و وضعیت خروجی‌ها آن را بعد از تغییر حالت با $Q_{(t)}$ و $\bar{Q}_{(t)}$ نشان دهیم، تحلیل فوق را می‌توانیم به صورت جدول ۲۱ خلاصه کنیم.

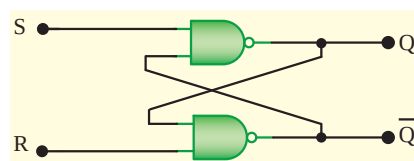
نکته



■ **فلیپ‌فلاپ SR با استفاده از گیت NAND:** فلیپ‌فلاپ SR را می‌توان با دروازه‌های NAND نیز طراحی کرد. در شکل ۴۷ مدار یک فلیپ‌فلاپ که با دروازه‌های NAND طراحی شده است را مشاهده می‌کنید، در جدول ۲۲ صحت آن دیده می‌شود.

جدول ۲۲- جدول صحت فلیپ‌فلاپ SR با گیت NAND

S	R	$Q_{(t)}$	
0	0	*	← غیر مجاز
0	1	1	← Set
1	0	0	← Reset
1	1	$Q_{(t-1)}$	← حالت ذخیره



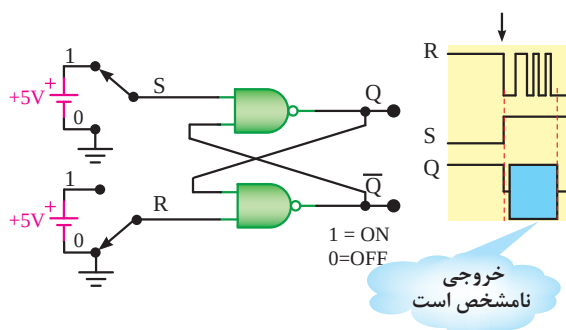
شکل ۴۷- فلیپ‌فلاپ SR با گیت‌های NAND



جدول صحت ۴ حالتی مدرا شکل ۴۷ را به دست آورید و آن را با جدول صحت ۴ حالتی فلیپ فلاپ طراحی شده با دروازه‌های NOR مقایسه کنید.

یکی از کاربردهای مدار فلیپ فلاپ SR حذف لرزش کلیدهای مکانیکی (Bounce) است. وقتی یک کلید، تغییر وضعیت می‌دهد به دلیل لرزش ناشی از این تغییر وضعیت، در طی چند میلی ثانیه اول قطع و وصل‌های مکرر در بین کنتاکت ثابت و کنتاکت متحرک کلید پدید آمده است، شکل ۴۸ وضعیت اثر لرزش کلیدها را نشان می‌دهد.

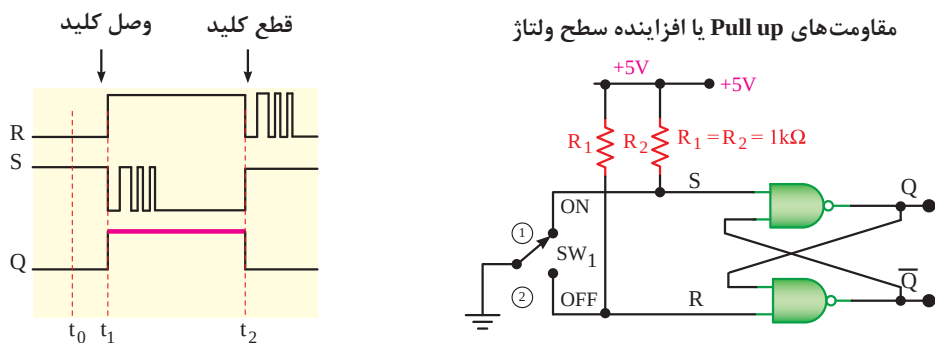
این قطع و وصل‌ها ممکن است باعث بروز اشتباهاتی در عملکرد مدار منطقی ترتیبی شود.



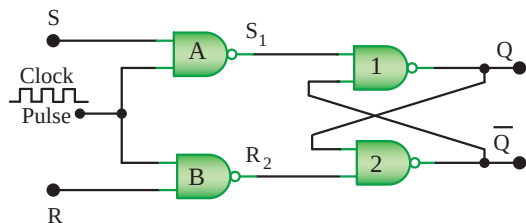
شکل ۴۸- اثر لرزش کلید در مدار فلیپ فلاپ SR

برای حذف این لرزش‌ها از مدار شکل ۴۹ می‌توان استفاده کرد. در این مدار اگر کلید را در حالت ۱ فرض کنیم فلیپ فلاپ Set می‌شود. حال اگر کلید تغییر وضعیت دهد، به محض اولین اتصال با کنتاکت ۲ مقدار $S=1$ می‌شود و فلیپ فلاپ را Reset می‌کند.

جدا شدن کنتاکت متحرک کلید از کنتاکت ثابت ۲، در فلیپ فلاپ حالت ذخیره ($R=S=1$) به وجود می‌آید. لذا خروجی‌های آن تغییر وضعیت نمی‌دهد. همین‌طور به هنگام تغییر وضعیت از ۲ به ۱ نیز با اولین تماس کنتاکت متحرک با کنتاکت ثابت ۱ فلیپ فلاپ Set می‌شود.



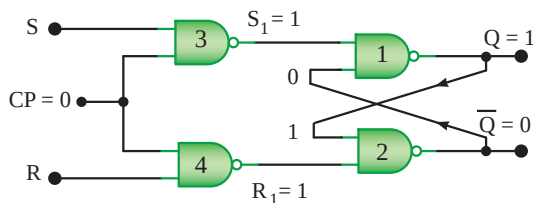
شکل ۴۹- مدار فلیپ فلاپ SR بدون اثر لرزش کلید



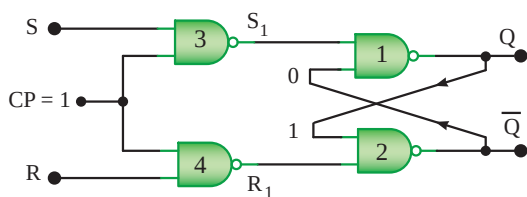
شکل ۵۰- مدار فلیپ فلاپ SR

جدول ۲۳- جدول صحت فلیپ فلاپ SR ساعتی

Clock	S	R	Q
0	0	0	تغییر نمی کند
0	0	1	
0	1	0	
0	1	1	
1	0	0	تغییر نمی کند
1	0	1	0 Reset
1	1	0	1 Set
1	1	1	غیرمجاز



شکل ۵۱- خروجی Q در مدار فلیپ فلاپ SR ساعتی حالت قبل را حفظ می کند و تغییر نمی کند.



شکل ۵۲- CP=1 و Q و Q-bar از مقادیر S و R تبعیت می کنند.

■ فلیپ فلاپ SR ساعتی

فلیپ فلاپ SR دو اشکال اساسی دارد:

۱ تغییر وضعیت سریع خروجی ها نسبت به تغییر وضعیت ورودی ها

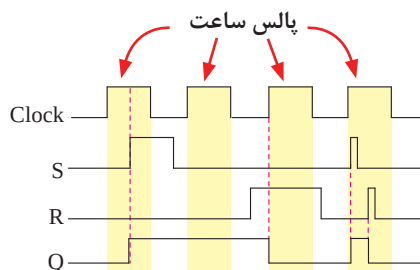
۲ وجود حالت تعریف نشده (غیرمجاز)

نخستین اشکال، تغییر وضعیت فوری خروجی ها نسبت به تغییر وضعیت ورودی ها است. یعنی به محض آنکه ورودی های مدار تغییر کند، خروجی های آن نیز متناسب با این تغییر وضعیت ها، تغییر می کند. برای رفع این مشکل، مدار فلیپ فلاپ SR را مطابق شکل ۵۰ اصلاح می کنیم. در این مدار پالس ساعت (Clock Pulse) به عنوان یک سیگنال فعال ساز عمل می کند.

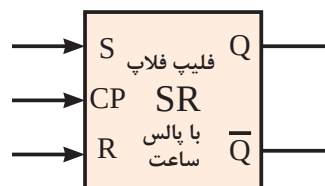
جدول ۲۳ جدول صحت فلیپ فلاپ SR ساعتی را نشان می دهد. Clock Pulse یا پالس ساعت ورودی (CP) به عنوان یک سیگنال فعال ساز عمل می کند. هنگامی که $CP = 0$ است، و شرایط S و R یک یا صفر باشد، در این حالت چون S_1 و R_1 همواره یک خواهند بود. مطابق با جدول ۲۳ (جدول صحت فلیپ فلاپ SR با گیت NAND) خروجی Q و Q-bar تغییر وضعیت نمی دهند و حالت قبل خود را حفظ می کنند، شکل ۵۱ نشان می دهد که وضعیت S و R هرچه باشد چون پالس ساعت در وضعیت صفر منطقی است، خروجی حالت قبل خود را حفظ می کند و ثابت می ماند.

اگر $CP = 1$ شود خروجی گیت NAND شماره ۳ و ۴ براساس ورودی های S و R تغییر می کند. در این حالت خروجی Q و Q-bar مطابق جدول درستی فلیپ فلاپ SR می شود. شکل ۵۲ مدار فلیپ فلاپ SR ساعتی را در زمان اعمال پالس ساعت نشان می دهد.

شکل ۵۳ بلوک دیاگرام فلیپ‌فلاپ SR ساعتی را نشان می‌دهد. در شکل ۵۴ نمودار زمانی ورودی‌ها و خروجی مدار فلیپ‌فلاپ SR ساعتی را مشاهده می‌کنید. ورودی‌های S و R فقط در زمان حضور Clock می‌توانند مؤثر واقع شوند. شکل موج ورودی‌ها و خروجی Q در نمودار زمانی مدار با توجه به جدول صحت فلیپ‌فلاپ SR ساعتی مطابق جدول ۲۳ رسم شده است.



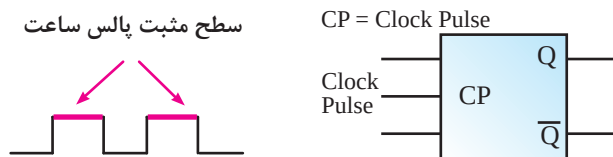
شکل ۵۴- نمودار زمانی ورودی‌ها و خروجی فلیپ‌فلاپ SR ساعتی



شکل ۵۳- بلوک دیاگرام فلیپ‌فلاپ SR ساعتی

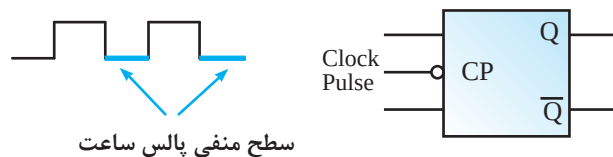
■ تقسیم‌بندی فلیپ‌فلاپ‌ها براساس پالس ساعت:

چگونگی عملکرد فلیپ‌فلاپ‌ها را براساس پالس ساعت نیز تقسیم‌بندی می‌کنند. در این قسمت‌بندی چهار حالت به‌وجود می‌آید.



شکل ۵۵- فلیپ‌فلاپ عامل با سطح مثبت پالس ساعت

■ **حالت اول:** عامل با سطح مثبت پالس ساعت اگر فلیپ‌فلاپ با سطح مثبت پالس ساعت عمل کند (عامل با سطح مثبت)، یعنی در شرایطی که Clock در حالت پایدار یک است، اطلاعات ورودی را بپذیرد آن را به‌صورت شکل ۵۵ نشان می‌دهند.



شکل ۵۶- فلیپ‌فلاپ عامل با سطح منفی پالس ساعت

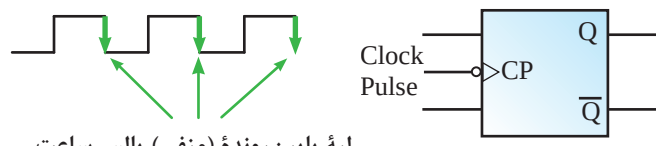
■ **حالت دوم:** عامل با سطح منفی پالس ساعت چنانچه فلیپ‌فلاپ با سطح منفی پالس ساعت عمل کند (عامل با سطح منفی)، در شرایطی که Clock Pulse در حالت پایدار صفر است و اطلاعات ورودی را بپذیرد آن را به‌صورت شکل ۵۶ نشان می‌دهند.



لبه بالا رونده پالس ساعت

شکل ۵۷- فلیپ فلاپ عامل با لبه بالا رونده پالس ساعت

■ **حالت سوم:** عامل با لبه بالا رونده پالس ساعت اگر فلیپ فلاپ با لبه بالا رونده (مثبت) پالس ساعت عمل کند (عامل با لبه بالا رونده)، یعنی در شرایطی که پالس ساعت لبه بالا رونده را طی می کند، اطلاعات را بپذیرد، آن را به صورت شکل ۵۷ نشان می دهند.



لبه پایین رونده (منفی) پالس ساعت

شکل ۵۸- فلیپ فلاپ عامل با لبه پایین رونده پالس

■ **حالت چهارم:** عامل با لبه پایین رونده پالس ساعت اگر فلیپ فلاپ با لبه پایین رونده (منفی) پالس ساعت عمل کند (عامل با لبه پایین رونده)، یعنی در شرایطی که پالس ساعت لبه پایین رونده را طی می کند، اطلاعات را بپذیرد آن را به صورت شکل ۵۸ نمایش می دهند.

بستن مدارهای ترتیبی (فلیپ فلاپ و تحقیق جدول صحت آن) به صورت نرم افزاری

هدف: کسب مهارت در شناخت عملکرد مدارهای ترتیبی

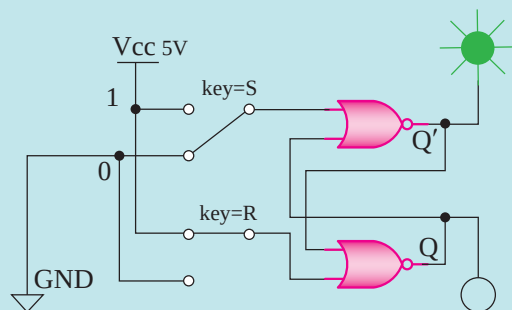
(مدار فلیپ فلاپ SR را با گیت های NOR و NAND فلیپ فلاپ SR ساعتی)

مواد، ابزار و تجهیزات: لوازم التحریر - اینترنت - رایانه - نرم افزار مولتی سیم یا هر نرم افزار مناسب دیگر
■ برای ذخیره اطلاعات دودویی در مدارهای دیجیتالی از حافظه ها استفاده می کنند. مدارهای ترتیبی از عناصر اصلی حافظه ها به شمار می روند. ساده ترین مدارهای ترتیبی فلیپ فلاپ ها هستند، که دو وضعیت منطقی پایدار «۰» و «۱» دارند. یکی از این نوع مدارها، فلیپ فلاپ SR است.

مراحل اجرای کار:

■ فلیپ فلاپ SR با دروازه های NOR

۱ در مدار فلیپ فلاپ با دروازه های NOR خروجی یک گیت به ورودی گیت دیگر فیدبک شده است. مدار دارای دو ورودی S و R و دو ورودی فیدبک Q و Q-bar است. برای تعیین وضعیت منطقی فعلی (زمان حال) فلیپ فلاپ، لازم است وضعیت ورودی S و R فیدبک Q و Q-bar در زمان قبل و حال را در نظر بگیریم. مدار فلیپ فلاپ SR با گیت NOR را در شکل ۵۹ مشاهده می کنید.



شکل ۵۹- مدار فلیپ فلاپ SR با گیت های NOR

کار عملی ۱



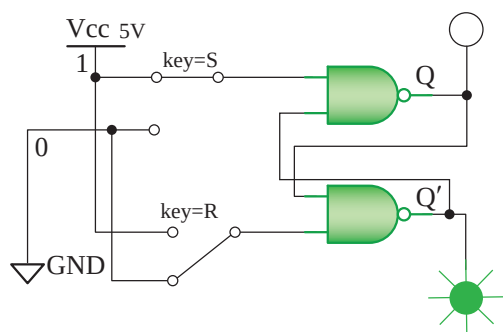
جدول ۲۴- جدول صحت فلیپ‌فلاپ SR با دروازه‌های NOR

S	R	$Q_{(t-1)}$	$Q_{(t)}$
0	0		
0	1		
1	0		
1	1		

۲ مدار فلیپ‌فلاپ شکل ۵۹ SR را بر روی میز کار نرم‌افزار مولتی‌سیم ببندید.

۳ با تغییر کلیدهای S و R مطابق جدول ۲۴ وضعیت منطقی خروجی را تعیین کرده و در جدول یادداشت کنید.

سؤال ۱: وضعیت ناپایدار این فلیپ‌فلاپ کدام حالت است؟ شرح دهید.



شکل ۶۰- مدار فلیپ‌فلاپ SR با دروازه‌های NAND

فلیپ‌فلاپ SR با دروازه‌های NAND

۱ مدار فلیپ‌فلاپ SR با دروازه‌های NAND را در شکل ۶۰ مشاهده می‌کنید.

جدول ۲۵- جدول صحت فلیپ‌فلاپ SR با دروازه‌های NAND

S	R	$Q_{(t-1)}$	$Q_{(t)}$
0	0		
0	1		
1	0		
1	1		

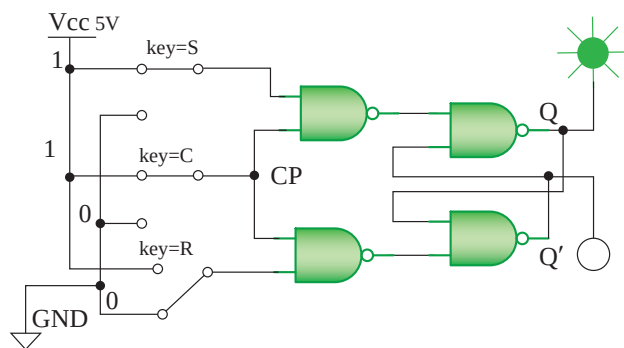
۲ مدار شکل ۶۰ را در فضای نرم‌افزار مولتی‌سیم پیاده‌سازی کنید.

۳ ورودی‌های set و Reset را به ترتیب و طبق جدول ۲۵ تغییر دهید و خروجی‌های مدار را مشاهده کنید و وضعیت منطقی آنها را بنویسید.

سؤال ۲: وضعیت ناپایدار این فلیپ‌فلاپ را شرح دهید.

سؤال ۳: جدول صحت این دو فلیپ‌فلاپ را با هم مقایسه کنید و تفاوت آن دو را بنویسید.

سؤال ۴: تغییرات آنی ورودی set و Reset در فلیپ‌فلاپ SR چه اشکالی به وجود می‌آورد؟ شرح دهید.



شکل ۶۱ - مدار فلیپ‌فلاپ SR ساعتی

■ فلیپ‌فلاپ SR ساعتی

۱ برای رفع اشکال عکس‌العمل فوری خروجی‌ها نسبت به ورودی‌ها در فلیپ‌فلاپ، می‌توانیم از ورودی پالس ساعت (Clock Pulse = cp) به عنوان یک سیگنال فعال‌ساز استفاده کنیم. هرگاه ورودی خط cp سطح ۵ ولت مثبت به عنوان یک پالس «۱» قرار گیرد، ورودی‌های S و R اجازه ورودی به مدار فلیپ‌فلاپ را پیدا می‌کنند و خروجی‌های فلیپ‌فلاپ تغییر وضعیت می‌دهند. مدار عملی این فلیپ‌فلاپ SR ساعتی در شکل ۶۱ نشان داده شده است.

جدول ۲۶ - جدول صحت فلیپ‌فلاپ SR ساعتی

cp	S	R	$Q_{(t-1)}$	$Q_{(t)}$
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

۲ مدار فلیپ‌فلاپ SR ساعتی شکل ۶۰ را ببندید.

۳ کلیدهای ورودی را به ترتیب جدول ۲۶ فلیپ‌فلاپ تغییر دهید، خروجی‌ها را مشاهده کنید و نتیجه را در جدول ۲۶ بنویسید.

سؤال ۵: با توجه به جدول ۲۶ در فلیپ‌فلاپ SR ساعتی خروجی‌ها در چه شرایطی تغییر وضعیت منطقی می‌دهند؟

سؤال ۶: آیا حالت غیرمجاز فلیپ‌فلاپ در این مدار بر طرف شده است؟ علت را توضیح دهید.

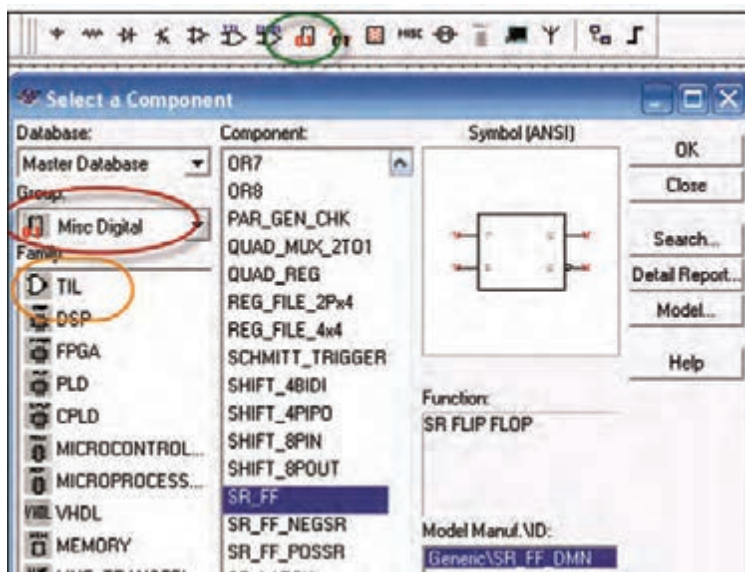


شکل ۶۲ - نماد بلوکی SR-FF

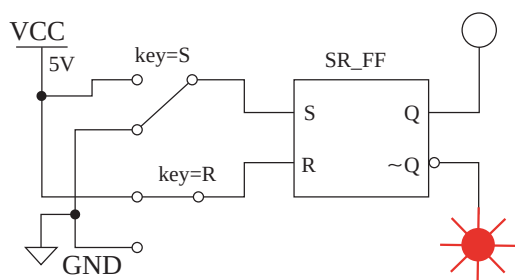
■ فلیپ‌فلاپ SR با نماد بلوکی یا IC

۱ فلیپ‌فلاپ SR را در مدارهای دیجیتالی با نماد بلوکی (IC) مطابق شکل ۶۲ نشان می‌دهند.

این نماد بلوکی در کتابخانه نرم‌افزار مولتی‌سیم شبیه‌سازی شده است و به عنوان یک مدار بلوکی و مشابه آی‌سی عمل می‌کند. این بلوک را از نوار قطعات (Component)، گروه Misc Digital و خانواده TIL انتخاب کنید. آدرس و مسیر انتخاب این فلیپ‌فلاپ و سایر عناصر مربوط به مدارهای منطقی را در شکل ۶۳ مشاهده می‌کنید.



شکل ۶۳- مسیر دستیابی به عناصر مربوط به مدارهای شبیه‌سازی شده دیجیتال



۲ مدار شکل ۶۴ که یک فلیپ‌فلاپ SR است را در فضای نرم‌افزاری ببینید.

شکل ۶۴- مدار فلیپ‌فلاپ SR با نماد بلوکی (آی‌سی)

جدول ۲۷- صحت فلیپ‌فلاپ SR بلوکی

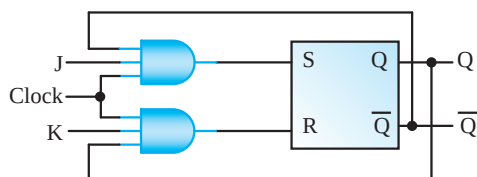
S	R	$Q_{(t-1)}$	$Q_{(t)}$
0	0		
0	1		
1	0		
1	1		

۲ با تغییر کلیدهای ورودی S و R جدول ۲۷ را کامل کنید.

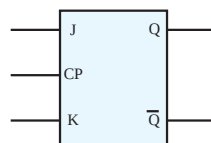
سؤال ۷: حالت غیرمجاز این فلیپ فلاپ را توضیح دهید.

■ فلیپ فلاپ J-K:

یکی از اشکال‌های فلیپ فلاپ S-R مربوط به حالت تعریف نشده آن یعنی وضعیت $S=R=1$ است. زیرا در این حالت وضعیت غیرمجاز پیش می‌آید. برای اصلاح این حالت از فلیپ فلاپ J-K استفاده می‌شود. در شکل ۶۵-ب مدار فلیپ فلاپ J-K رسم شده است. نماد بلوکی فلیپ فلاپ J-K را در شکل ۶۵-الف مشاهده می‌کنید.



ب) فلیپ فلاپ J-K



الف) نماد بلوک دیاگرام فلیپ فلاپ J-K

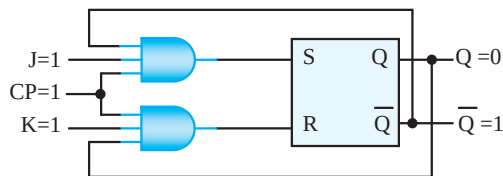
شکل ۶۵

جدول صحت فلیپ فلاپ J-K در جدول ۲۸ آمده است. این جدول نشان می‌دهد حالت ممنوعه برطرف شده است یعنی هنگامی که $J=K=1$ می‌شود، اگر فلیپ فلاپ مطابق شکل ۶۶ در حالت Set یعنی $Q=1$ قرار داشته باشد، وضعیت آن تغییر می‌کند و Reset می‌شود.

جدول ۲۸- صحت فلیپ فلاپ J-K

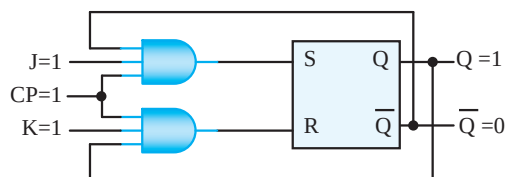
ورودی J	ورودی K	خروجی بعد از اعمال پالس ساعت
J	K	$Q_{(t)}$
0	0	$Q_{(t-1)}$
0	1	0
1	0	1
1	1	$\bar{Q}_{(t-1)}$
خروجی قبل از اعمال پالس ساعت		$Q_{(t-1)}$

۱ فیدبک می‌یابد.



۰ فیدبک می‌یابد.

الف)



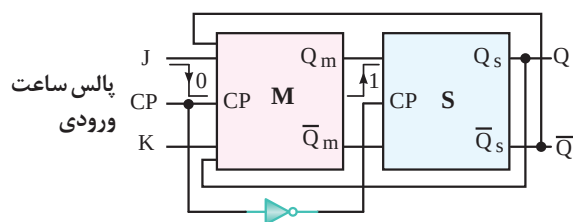
ب) با فیدبک خروجی Q و $\bar{Q}$ به ورودی، وضعیت جدید $Q=1$ و $\bar{Q}=0$ ظاهر شده است.

شکل ۶۶

همچنین اگر فلیپ‌فلاپ در وضعیت Reset یعنی $Q=0$ باشد با برقراری $J=K=1$ به وضعیت Set می‌رود یعنی $Q=1$ می‌شود به عبارت دیگر حافظه به حالتی برعکس وضعیت قبلی خود تغییر حالت می‌دهد. این وضعیت را که شبیه قطع و وصل کردن یک کلید است حالت کلیدی می‌نامند. باید توجه داشت تغییر وضعیت Q و $\bar{Q}$ زمانی رخ می‌دهد که پالس ساعت فعال باشد، ($CP=1$) طبیعی است اگر پالس ساعت فعال نباشد یعنی $CP=0$ Q و $\bar{Q}$ تغییر وضعیت نمی‌دهند و حالت اولیه خود را حفظ می‌کنند.

■ **عیب فلیپ‌فلاپ J-K**: همان‌طور که مشاهده کردید فلیپ‌فلاپ J-K توانسته است حالت تعریف نشده فلیپ‌فلاپ S-R را برطرف کند. چون خروجی‌های Q و $\bar{Q}$ مستقیماً به ورودی‌ها فیدبک شده‌اند. اگر در این حالت پالس ساعت برابر با (1) باقی بماند، به دلیل وجود فیدبک، مقادیر مرتباً تغییر می‌کنند و خروجی فلیپ‌فلاپ دائماً بین صفر و یک نوسان می‌کند.

در شکل ۶۶ الف و ب تغییر وضعیت Q و $\bar{Q}$ نشان داده شده است. وضعیت جدید $Q=1$ و $\bar{Q}=0$ ظاهر شده است. این حالت فلیپ‌فلاپ را پدیده دور خود چرخیدن (Race Around) می‌گویند. اگر پالس ساعت مساوی صفر ($CP=0$) شود، در این لحظه Q روی آخرین وضعیت خود ثابت می‌ماند. برای برطرف کردن این عیب از فلیپ‌فلاپ JK-MS استفاده می‌شود.

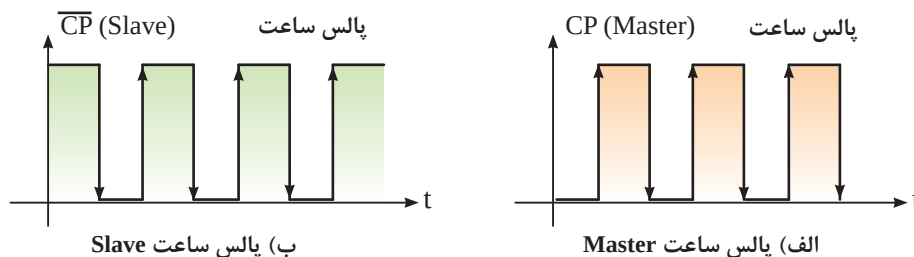


شکل ۶۷- فلیپ‌فلاپ JK-

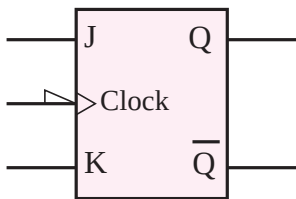
■ فلیپ‌فلاپ JK-MS (JK-Master slave):

فلیپ‌فلاپ JK-MS از دو فلیپ‌فلاپ S-R مجزا درست شده است که یکی (اصلی فرمانده - Master) و دیگری (پیرو - فرمانبر - Slave) نام دارد. شکل ۶۷ شکل بلوکی این فلیپ‌فلاپ را نشان می‌دهد. خطوط فیدبک از Q و $\bar{Q}$ به ورودی‌ها رسم شده است.

با توجه به بلوک شکل ۶۸ درمی‌یابیم که وقتی پالس ساعت فلیپ‌فلاپ Master برابر با NOT می‌شود به عنوان پالس ساعت فلیپ‌فلاپ Slave عمل می‌کند. شکل‌های ۶۸ الف و ب وضعیت پالس ساعت Master و Slave را نسبت به هم نشان می‌دهد.



شکل ۶۸- وضعیت پالس ساعت فلیپ‌فلاپ Master و Slave



شکل ۶۹- بلوک دیاگرام فلیپ فلاپ JK-MS

جدول ۲۹- صحت فلیپ فلاپ JK-

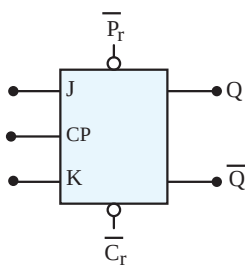
J	K	CP	Q_t
0	0		$Q_{(t-1)}$ حالت قبلی را حفظ می‌کند
0	1		0 RESET
1	0		1 SET
1	1		$\bar{Q}_{(t-1)}$ حالت قبلی عکس می‌شود

هنگامی که پالس از ولتاژ صفر به سمت سطح ولتاژ زیاد تغییر می‌کند، به این بخش لبه صعودی پالس بالارونده گویند. شکل ۶۸- الف لبه صعودی پالس را نشان می‌دهد. هنگامی که پالس از سطح ولتاژ زیاد به سمت سطح ولتاژ صفر نزول می‌کند این بخش، لبه نزولی پالس (پایین رونده) نام دارد. شکل ۶۸- ب لبه نزولی پالس را نشان می‌دهد.

در فلیپ فلاپ JK-MS در هنگام لبه صعودی پالس Master، اطلاعات ورودی وارد حافظه Master Q می‌شود. شکل ۶۹ نشان می‌دهد که اطلاعات ورودی وارد حافظه Master شده است. جدول درستی فلیپ فلاپ JK-MS مانند جدول درستی فلیپ فلاپ JK است. فقط در فلیپ فلاپ JK-MS اطلاعات در لبه نزولی پالس ساعت وارد حافظه می‌شوند.

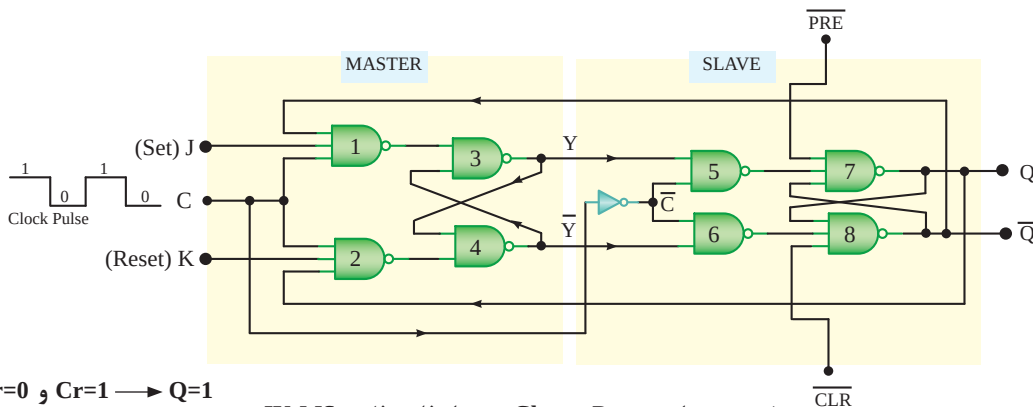
جدول ۲۹ صحت فلیپ فلاپ JK-MS را نشان می‌دهد. بلوک دیاگرام فلیپ فلاپ JK-MS را در شکل ۶۹ ملاحظه می‌کنید.

■ تنظیم عملکرد ورودی‌های پاک کردن (Clear) و پیش تنظیم (Preset) Pre:



الف) بلوک دیاگرام فلیپ فلاپ JK با Pr و Clr

در زمان وصل تغذیه به مدار فلیپ فلاپ، وضعیتی که حافظه در آن قرار می‌گیرد یعنی حالت Q، در انواع فلیپ فلاپ‌ها کاملاً تصادفی است. برای ایجاد یک حالت معین و تعریف شده اولیه از ورودی پیش تنظیم (Preset) و پاک کردن (Clear) استفاده می‌کنیم. در شکل ۷۰- الف بلوک دیاگرام فلیپ فلاپ JK با Pr و Clr نشان داده شده است. Pre و Clr مانند شکل ۷۰- ب مستقیماً روی Q و Q-bar اثر می‌گذارند. در صورتی که $Pr = 0$ و $Cr = 1$ باشد مقدار $Q = 1$ می‌شود، همچنین اگر $Pr = 1$ و $Cr = 0$ باشد مقدار $Q = 0$ خواهد شد.



$Pr=0$ و $Cr=1 \rightarrow Q=1$

$Pr=1$ و $Cr=0 \rightarrow Q=0$

ب) ورودی‌های Clear و Preset در مدار فلیپ فلاپ JK-MS

شکل ۷۰

جدول ۳۰- فلیپ‌فلاپ JK با ورودی‌های Pr و Clr

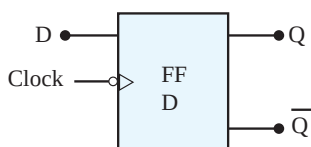
P_r	C_r	CP	J	K	Q_t
0	0	X	X	X	* حالت ناپایدار
0	1	X	X	X	1 پر شده Preset
1	0	X	X	X	0 پاک شده Clear
1	1	—	X	X	حالت قبلی را حفظ می‌کند $Q_{(t-1)}$
1	1		0	0	$Q_{(t-1)}$
1	1		0	1	0 Reset
1	1		1	0	1 Set
1	1		1	1	حالت قبلی عکس می‌شود $\overline{Q}_{(t-1)}$

ایجاد شرایط $Pr = 0$ و $Cr = 0$ حالت ناپایدار ایجاد می‌کند. وضعیت $Pr = 1$ و $Cr = 1$ تغییری در حافظه به وجود نمی‌آورد.

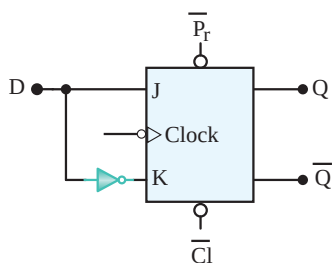
در شکل ۷۵ یک فلیپ‌فلاپ JK همراه با خط‌های Pr و Cr رسم شده است.

■ فلیپ‌فلاپ نوع D (تأخیری یا Delay):

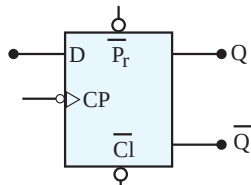
این فلیپ‌فلاپ تنها دارای یک ورودی است که ورودی (D) نام دارد. (شکل ۷۱). از این فلیپ‌فلاپ به عنوان یک سلول ثبات (ثبت‌کننده) استفاده می‌کنند. زیرا اطلاعات ورودی آن پس از هر پالس ساعت وارد حافظه فلیپ‌فلاپ می‌شود و تا پالس ساعت بعدی آن را حفظ می‌کند.



شکل ۷۱- فلیپ‌فلاپ D



الف) فلیپ‌فلاپ D با استفاده از فلیپ‌فلاپ JK



ب) بلوک دیگرام فلیپ‌فلاپ D

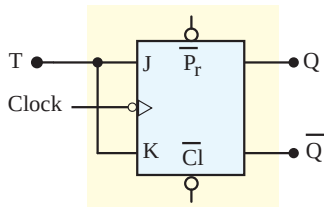
شکل ۷۲

جدول ۳۱- جدول صحت فلیپ‌فلاپ D

CP	D	Q
	0	0
	1	1

جدول درستی فلیپ‌فلاپ D مطابق جدول ۳۱ است. در شکل ۷۲ بلوک دیگرام فلیپ‌فلاپ نوع D رسم شده است.

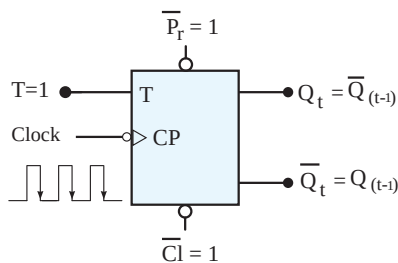
■ فلیپ فلاپ نوع T (کلیدی یا Toggle):



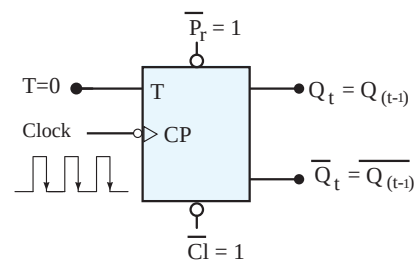
اگر دو ورودی فلیپ فلاپ JK را به هم وصل کنیم و اتصال مشترک را T بنامیم، فلیپ فلاپ جدیدی ساخته می‌شود که به آن فلیپ فلاپ نوع T یا کلیدی گویند. شکل ۷۳ نحوه تبدیل فلیپ فلاپ J-K را به T نشان می‌دهد. با توجه به شکل درمی‌یابیم که اطلاعات ورودی‌های J و K با هم برابرند.

شکل ۷۳- تبدیل فلیپ فلاپ J-K به T

پس اگر $T = 0$ باشد، مطابق شکل ۷۴، $J = K = 0$ است. در این صورت حافظه بدون تغییر می‌ماند. حال چنانچه $T = 1$ باشد، $J = K = 1$ است. در این حالت طبق شکل ۷۵ با هر پالس ساعت وضعیت حافظه برعکس حالت قبل می‌شود. یعنی اگر حافظه Set است به حالت Reset و اگر Reset است به حالت Set تغییر حالت می‌دهد. جدول ۳۲ جدول درستی T-FF بلوک دیاگرام فلیپ فلاپ T در شکل ۷۷ آمده است.



شکل ۷۵، $T = 1$ است در لبه نزولی پالس ساعت Q_t برعکس می‌شود.



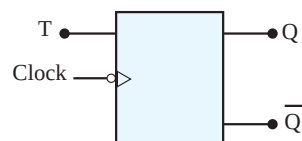
شکل ۷۴، $T = 0$ است و Q و $\bar{Q}$ تغییر نمی‌کنند.

جدول ۳۲- درستی T-FF

T	CP	Q_t
0		$Q_{(t-1)}$
1		$\bar{Q}_{(t-1)}$

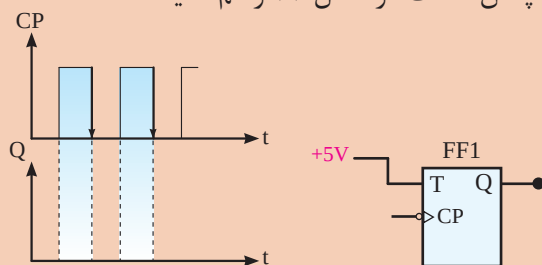
حالت قبلی را حفظ می‌کند

حالت قبلی عکس می‌شود.



شکل ۷۶- بلوک دیاگرام T-FF

شکل موج خروجی Q فلیپ فلاپ T را به ازای دو پالس ساعت در شکل ۷۷ رسم کنید.



شکل ۷۷- T-FF

تمرین کلاسی

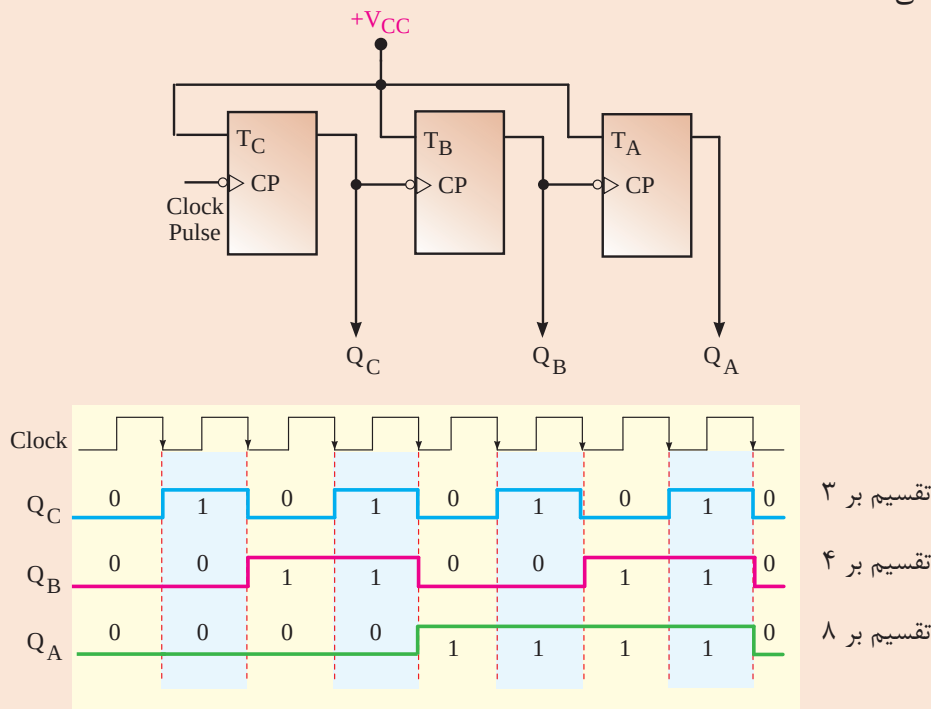


فکر کنید



■ کاربرد فلیپ فلاپ T:

از فلیپ فلاپ T اغلب به عنوان تقسیم کننده فرکانس در ساختمان مدارهای ساعت یا شمارنده استفاده می شود. شکل ۷۷ یک شمارنده به عنوان تقسیم کننده فرکانس را نشان می دهد. اگر سه طبقه فلیپ فلاپ را مطابق شکل ۷۸ به هم وصل کنیم، فرکانس و شکل موج خروجی هریک از فلیپ فلاپ ها نصف شکل موج پالس ورودی یا شکل موج خروجی فلیپ فلاپ ماقبل است. همان طور که مشاهده می کنید، فلیپ فلاپ طبقه C در حکم یک تقسیم کننده فرکانس پالس ساعت عمل می کند.



شکل ۷۸

سؤال: اگر فرکانس CP (Clock Pulse) برابر ۱۰۰ کیلوهرتز باشد؛ فرکانس Q_C و Q_B و Q_A چقدر است؟

کار عملی



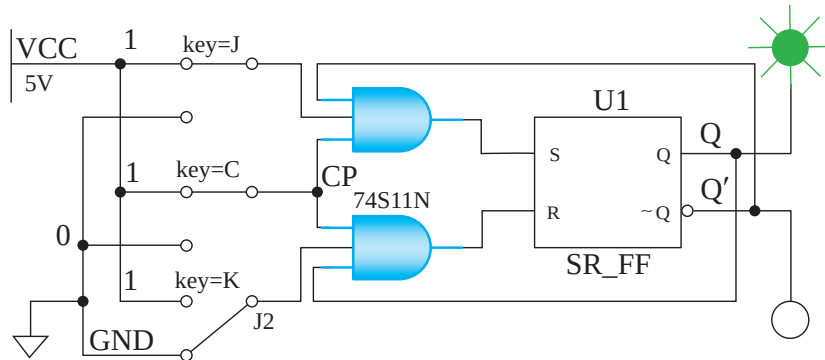
بستن مدارهای ترتیبی (فلیپ فلاپ ها و تحقیق جدول صحت آن) به صورت نرم افزاری

هدف: کسب مهارت در شناخت عملکرد مدارهای ترتیبی (مدار فلیپ فلاپ JK, D, T)

مواد، ابزار و تجهیزات: لوازم التحریر - اینترنت - رایانه - نرم افزار مولتی سیم یا هر نرم افزار مناسب دیگر
مراحل اجرای کار:

■ فلیپ فلاپ JK :

۱ برای رفع اشکال حالت غیرمجاز فلیپ فلاپ SR از فلیپ فلاپ JK استفاده می شود. ساختار اصلی این فلیپ فلاپ مطابق شکل ۷۹ است.



شکل ۷۹- مدار فلیپ فلاپ JK با ساختار اصلی

۲ مدار فلیپ فلاپ شکل ۷۹ را بر روی میز کار مجازی ببندید. با تغییر کلیدهای J، K و CP طبق جدول ۳۳ صحت وضعیت خروجی Q و $\bar{Q}$ را مشخص کنید و نتایج را در جدول بنویسید.

جدول ۳۳- صحت فلیپ فلاپ JK

CP	K	J	$Q_{(t-1)}$	$Q_{(t)}$
0	0	0		
0	0	1		
0	1	0		
0	1	1		
1	0	0		
1	0	1		
1	1	0		
1	1	1		

سؤال ۱: آیا حالت غیرمجاز فلیپ فلاپ SR در JK-FF برطرف شده است؟

.....

.....

سؤال ۲: وضعیت خروجی‌ها در حالت $CP = 1$, $J=K=1$ را در مدار فلیپ‌فلاپ JK شرح دهید.

.....

.....

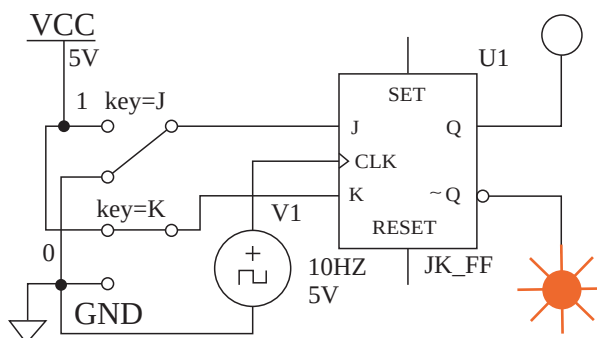
۳ نماد بلوکی فلیپ‌فلاپ JK را از گروه Misc Digital انتخاب کنید و بر روی میز کار بیاورید.

مدار فلیپ‌فلاپ JK را با استفاده از نماد بلوکی آن مطابق شکل ۸۰ ببندید.

با تغییر کلیدهای J، K و CP طبق جدول ۳۴ صحت وضعیت خروجی Q و $\bar{Q}$ را مشخص کنید و نتایج را در جدول بنویسید.

جدول ۳۴- جدول صحت فلیپ‌فلاپ JK

J	K	$Q(t)$
0	0	
0	1	
1	0	
1	1	



شکل ۸۰- مدار بلوکی فلیپ‌فلاپ JK

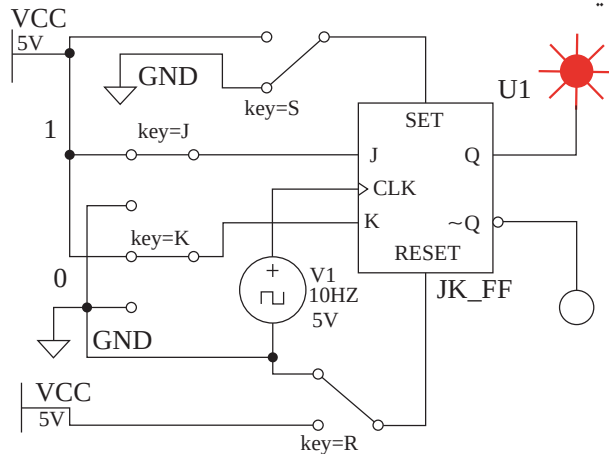
سؤال ۳: اگر در مدار شکل ۸۰ حالت $J=K=1$ برقرار باشد برای خروجی‌ها چه وضعیتی اتفاق می‌افتد؟ شرح دهید.

.....

.....

■ استفاده از Preset و Clear: در مدارهای فلیپ‌فلاپ، با روشن شدن منبع تغذیه، وضعیت‌های خروجی Q و $\bar{Q}$ به صورت تصادفی در وضعیت «۱» یا «۰» قرار می‌گیرند. برای آنکه خروجی‌ها به یک حالت معین و از پیش تنظیم شده در لحظه شروع کار مدار برسند، از دو ورودی مستقل Preset و Clear استفاده می‌کنند.

۱ مدار شکل ۸۱ را ببندید.



شکل ۸۱- مدار عملکرد ورودی‌های Preset و Clear

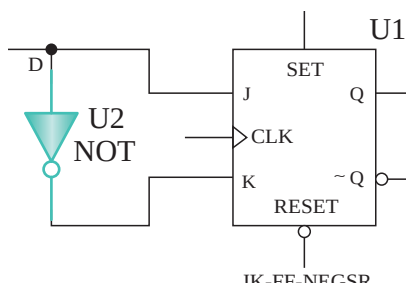
۲ در IC های فلیپ‌فلاپ، پایه Preset را با set و Clear را با Reset مشخص می‌کنند. ورودی‌های مدار شکل را مطابق جدول ۳۵ تغییر دهید. وضعیت خروجی‌های Q و $\bar{Q}$ را مشاهده کنید و در جدول بنویسید.

جدول ۳۵- صحت فلیپ‌فلاپ JK با ورودی‌های CL و Pr

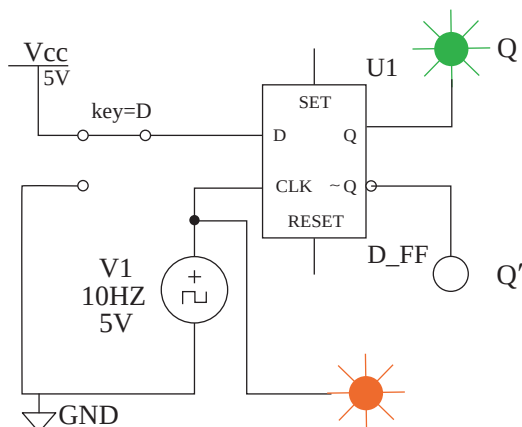
Pr=S	CL=R	CP	J	K	$Q_{(t)}$	$(t)\bar{Q}$
0	0	X	X	X		
0	1	X	X	X		
1	0	X	X	X		
1	1		X	X		
1	1		0	0		
1	1		0	1		
1	1		1	0		
1	1		1	1		

سؤال ۴: ورودی CL و Pr با کدام وضعیت منطقی «۱» و «۰» فعال می‌شوند؟ توضیح دهید.

سؤال ۵: حالت نامعین در جدول ۳۵ کدام حالت است؟ شرح دهید.



شکل ۸۲- فلیپ فلاپ D



شکل ۸۳- مدار فلیپ فلاپ D

■ فلیپ فلاپ D:

۱ فلیپ فلاپ D دارای یک ورودی است که در مدارهای دیجیتالی ترتیبی به عنوان ثبت کننده یک بیت باینری استفاده می شود. این فلیپ فلاپ را می توان مطابق شکل ۸۲ با استفاده از JK-FF طراحی کرد.

۲ فلیپ فلاپ D را از نوار Misc Digital و Component

انتخاب کنید و آن را بر روی میز کار بیاورید.

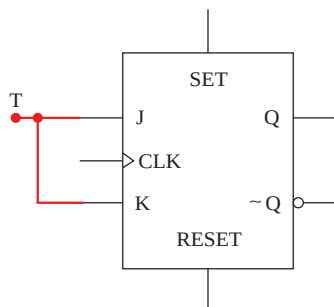
۳ مدار شکل ۸۳ را بر روی میز کار ببندید.

جدول ۳۶- جدول صحت فلیپ فلاپ D

CP	D	$Q(t)$	$(t) \overline{Q}$
	0		
	1		

۴ کلید ورودی D را تغییر وضعیت دهید و حالت خروجی را پس از مشاهده در جدول ۳۶ یادداشت کنید.

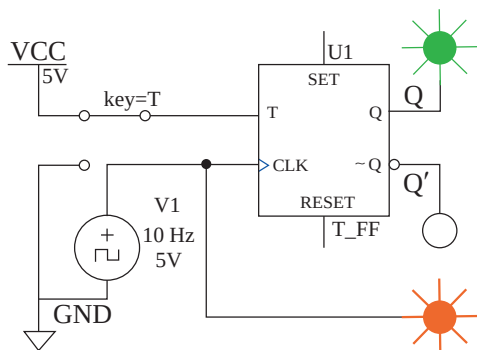
سؤال ۶: اگر $D = 1$ بعد از پنج پالس ساعت خروجی در چه وضعیتی قرار می گیرد؟ شرح دهید.



شکل ۸۴- فلیپ فلاپ T

■ فلیپ فلاپ T:

حرف T در فلیپ فلاپ نوع T ابتدای کلمه Toggle به معنی تغییر وضعیت است. در صورتی که دو ورودی J و K فلیپ فلاپ JK را به هم اتصال دهیم فلیپ فلاپ نوع T ساخته می شود (شکل ۸۴).



شکل ۸۵- مدار فلیپ فلاپ T

۱ فلیپ فلاپ T را طبق شکل ۸۵ بر روی میز کار نرم افزار ببندید.

جدول ۳۷- جدول صحت فلیپ فلاپ T

CP	D	Q(t)	$\overline{Q}(t)$
	0		
	1		

۲ با تغییر کلید T وضعیت خروجی ها را بررسی کنید و جدول ۳۷ را کامل نمایید.

سؤال ۷: چنانچه $T=1$ باشد، برای خروجی Q چه اتفاقی می افتد؟ این حالت چه نام دارد؟ شرح دهید.



■ کاربرد T-FF:

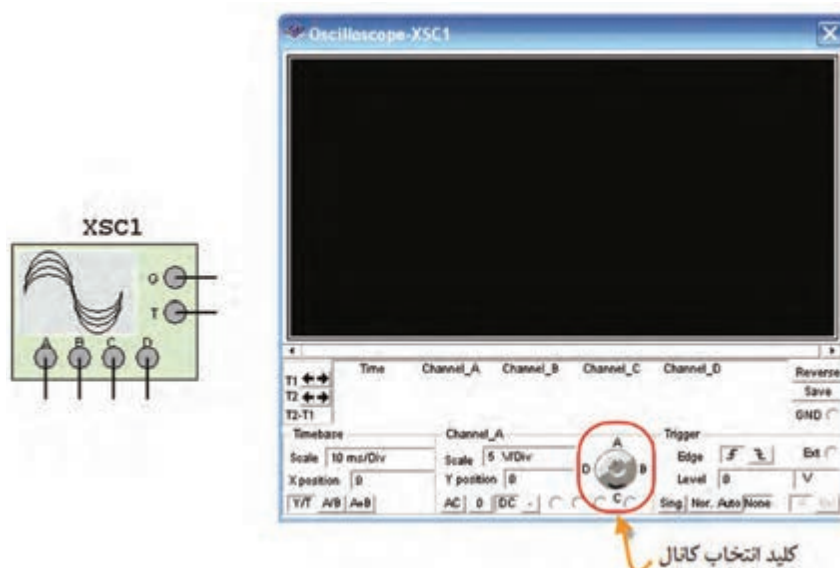
۱ در مدارهای دیجیتالی برای تقسیم فرکانس از T-FF استفاده می شود. برای انجام آزمایش تقسیم فرکانس نیاز به یک دستگاه اسیلوسکوپ چهار کاناله داریم.

۲ دستگاه اسیلوسکوپ چهار کاناله را از نوار Instruments بردارید و آن را مطابق شکل ۸۶ بر روی میز کار نرم افزار بیاورید.

شکل ۸۶- محل قرار گرفتن دستگاه اسیلوسکوپ چهار کاناله در نرم افزار

۳ در دستگاه اسیلوسکوپ چهار کاناله یک کلید انتخاب کانال های ورودی وجود دارد که با قرار دادن مکان نما روی سلکتور و کلیک کردن روی موس، کانال مورد نظر فعال می شود.

در این حالت Volt/Div (Scale)، Time/Div و تغییر مکان عمودی برای آن فعال می‌شود، کانال قابل تنظیم است، در شکل ۸۷ کلید انتخاب کانال نمایش داده شده است.



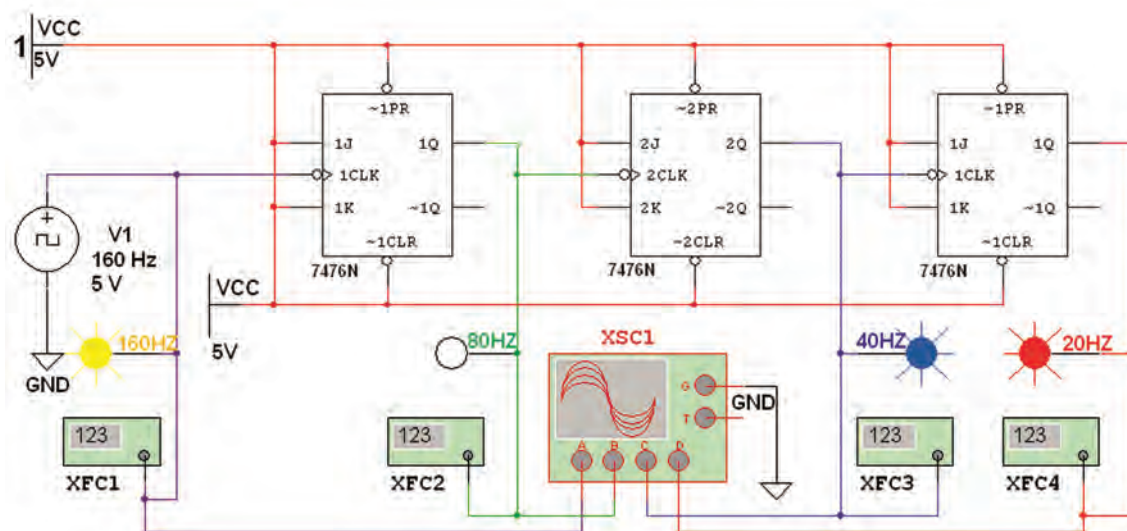
شکل ۸۷- کلید انتخاب کانال در اسیلوسکوپ چهارکاناله

۴ برای اندازه‌گیری فرکانس خروجی مدارهای الکترونیکی و دیجیتالی از دستگاه فرکانس‌متر استفاده می‌کنند. این دستگاه را می‌توانید مطابق شکل ۸۸ از نوار Instruments انتخاب کنید. این دستگاه یک ورودی دارد و اگر به ورودی یا خروجی مدار اتصال داده شود، فرکانس را اندازه می‌گیرد.



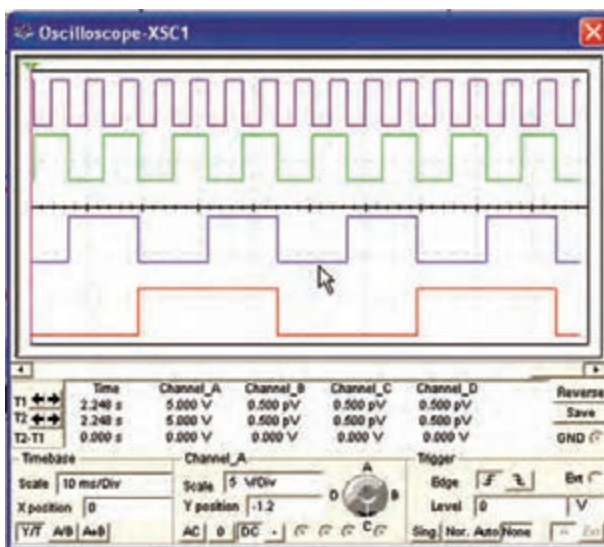
شکل ۸۸- موقعیت فرکانس‌متر در نوار ابزار

۵ مدار شکل ۸۹ را با دقت بر روی میز کار نرم افزار ببندید. فرکانس ورودی را روی ۱۶۰ هرتز تنظیم کنید.



شکل ۸۹- مدار تقسیم کننده فرکانس بر ۲، ۴ و ۸ با استفاده از فلیپ فلاپ نوع T

۶ مدار را راه اندازی کنید، باید لامپ های پروپ لاجیک، زرد، سبز، آبی و قرمز متناسب با فرکانس خاموش و روشن شود. بیشترین تغییر حالت خاموش و روشن مربوط به لامپ زرد (۱۶۰Hz) و کمترین تغییر مربوط به لامپ قرمز است. در صورتی که مدار راه اندازی نشد، یک بار پوشه مدار را ببندید و دوباره باز کنید. پس از راه اندازی مجدداً اگر باز هم راه اندازی نشد، نقشه مدار را مجدداً بازبینی کنید و اشکال آن را برطرف نمایید. پس از راه اندازی، اسیلوسکوپ را روشن کنید.



۷ ولوم تغییر مکان عمودی (Y Position) را برای هر کانال به گونه ای تنظیم کنید، تا به ترتیب شکل موج پالس های ۱۶۰Hz، ۸۰Hz، ۴۰Hz و ۲۰Hz را از بالا به پایین نشان دهد.

۸ با تنظیم کلید Time/Div (Time Base) برای هر شکل موج با استفاده از شکل ۹۰ فرکانس را اندازه گیری کنید.

شکل ۹۰- تنظیم اسیلوسکوپ چهار کاناله برای نمایش پالس ساعت ورودی و خروجی های مدار شکل ۸۹

۹ با کلیک کردن روی فرکانس مترها پالس ساعت ورودی و خروجی هریک از فلیپ‌فلاپ‌ها را مطابق شکل ۹۱ اندازه‌گیری کنید.



شکل ۹۱- اندازه‌گیری فرکانس پالس ساعت ورودی و فرکانس‌های خروجی فلیپ‌فلاپ‌های مدار شکل ۸۹ با استفاده از فرکانس‌متر

شیفت رجیسترها (Shift Registers)

✓ شیفت رجیسترها برای ثبت و حفظ اطلاعات باینری به صورت موقتی به کار می‌روند. علاوه بر این، می‌توان از آنها برای انتقال اطلاعات نیز استفاده کرد. شیفت رجیسترها به علت کاربرد وسیعی که در سیستم‌های دیجیتالی دارند با قابلیت‌های متفاوت و تنوع زیاد در بازار یافت می‌شوند. در جدول ۳۸ انواع شیفت رجیسترها از نظر چگونگی ورود و خروج اطلاعات (۰ و ۱ باینری) آورده شده است.

جدول ۳۸- انواع شیفت رجیستر

Serial input- Serial output (SISO)

ورودی سری - خروجی سری

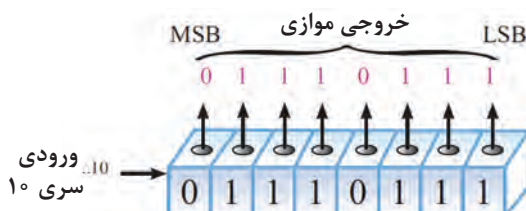
اطلاعات (بیت‌های) ورودی یکی پس از دیگری و به صورت سریال وارد اولین بیت فلیپ‌فلاپ می‌شوند و سپس از فلیپ‌فلاپ‌های بعدی عبور می‌کنند.



Serial input- Parallel output (SIPO)

ورودی سری - خروجی موازی

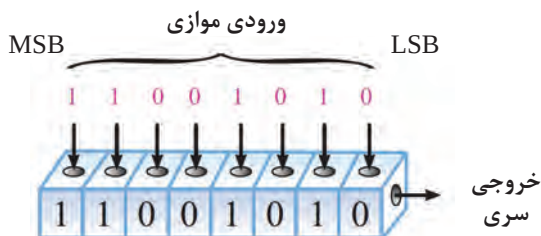
اطلاعات ورودی به صورت سریال وارد مدار می‌شود و خروجی‌ها به طور هم‌زمان و موازی دریافت می‌گردند.



Parallel input- Serial output (PISO)

ورودی موازی - خروجی سری

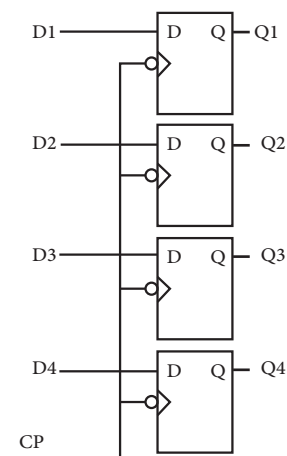
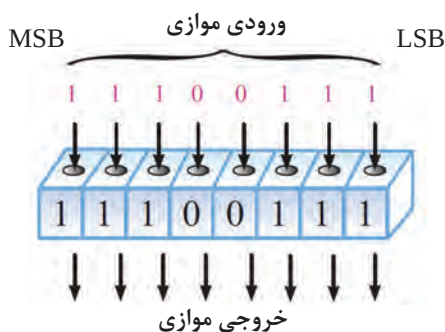
اطلاعات به صورت موازی وارد شیفت رجیستر شده و سری دریافت می شود.



Parallel input- Parallel output (PIPO)

ورودی موازی - خروجی موازی

اطلاعات به صورت موازی وارد شیفت رجیستر شده و سری دریافت می شود.



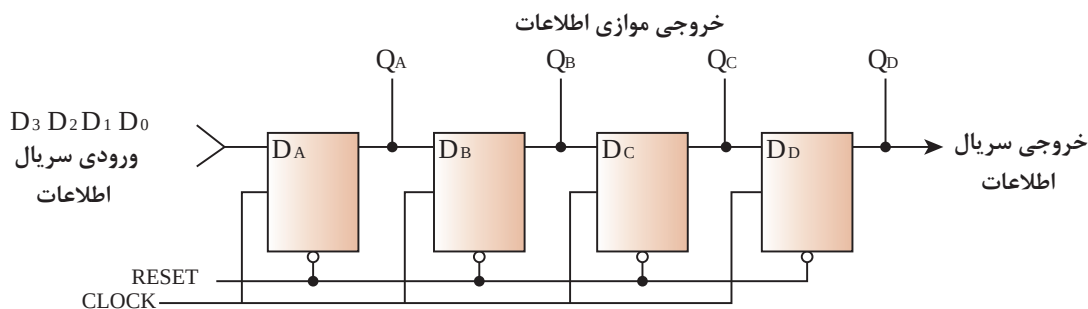
شکل ۹۲- شیفت رجیستر ورودی موازی و خروجی موازی

انتقال اطلاعات در شیفت رجیستر از طریق ورودی های اصلی فلیپ فلاپ D یا از ورودی های Preset و Clear صورت می گیرد. در شکل ۵۴ یک شیفت رجیستر با ورودی موازی و خروجی موازی نشان داده شده است که در آن انتقال اطلاعات از طریق ورودی های فلیپ فلاپ D انجام می شود.

در فلیپ فلاپ D با اعمال پالس ساعت، اطلاعات ورودی به خروجی انتقال می یابد. هنگامی که پالس ساعت $CP = 0$ است، اطلاعات خروجی Q_1 تا Q_4 تغییر پیدا نمی کند. در لبه نزولی پالس ساعت، اطلاعات D_1, D_2, D_3, D_4 به خروجی های Q_1, Q_2, Q_3, Q_4 انتقال می یابند.

شکل ۹۳ یک شیفت رجیستر با ورودی سری و خروجی سری (SISO) را نشان می دهد که می تواند به صورت شیفت رجیستر با ورودی سری و خروجی موازی (SIPO) هم عمل کند.

در این نوع رجیستر می‌توان اطلاعات ثبت شده را به‌طور هم‌زمان نیز از حافظه خارج کرد. ابتدا توسط خط Reset همه حافظه‌ها پاک می‌شود. با هر پالس ساعت، اطلاعات از طریق ورودی‌های سریال به‌صورت پشت سرهم وارد حافظه می‌شود و به سمت راست انتقال می‌یابد.

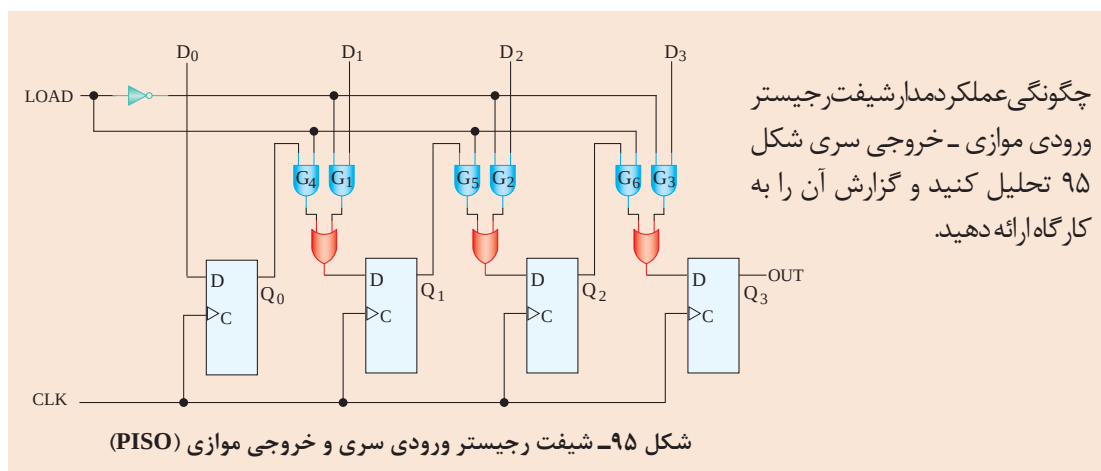


شکل ۹۳- شیفت رجیستر (SISO) یا (SIPO)

بعد از چهار پالس ساعت، اطلاعات در چهار حافظه ثبت می‌شود. شکل ۹۴ در این حالت می‌توان اطلاعات چهار فلیپ‌فلاپ را هم‌زمان نوشت (Q_A, Q_B, Q_C, Q_D) در این صورت مدار به‌عنوان یک شیفت رجیستر با ورودی سری و خروجی موازی (SIPO) عمل می‌کند. برای خروج سریال اطلاعات چهار پالس ساعت دیگر به مدار اعمال می‌شود، با دریافت اطلاعات از خروجی Q_D مدار عملکرد شیفت رجیستر با ورودی سری و خروجی سری (SISO) را دارد.

	Q_A	Q_B	Q_C	Q_D	
$D_4 D_3 D_2 D_1$	0	0	0	0	همه حافظه‌ها پاک
$D_4 D_3 D_2$	D_1				D_1 وارد حافظه شد
$D_4 D_3$	D_2	D_1			D_1, D_2 وارد حافظه شدند
D_4	D_3	D_2	D_1		D_1, D_2, D_3 وارد حافظه شدند
	D_4	D_3	D_2	D_1	D_1, D_2, D_3, D_4 وارد حافظه شدند

شکل ۹۴- ورود اطلاعات به‌صورت سریال یا پشت سر هم به شیفت رجیستر



شکل ۹۵- شیفت رجیستر ورودی سری و خروجی موازی (PISO)

بررسی کنید



چگونگی عملکرد مدار شیفت رجیستر ورودی موازی - خروجی سری شکل ۹۵ تحلیل کنید و گزارش آن را به کارگاه ارائه دهید.

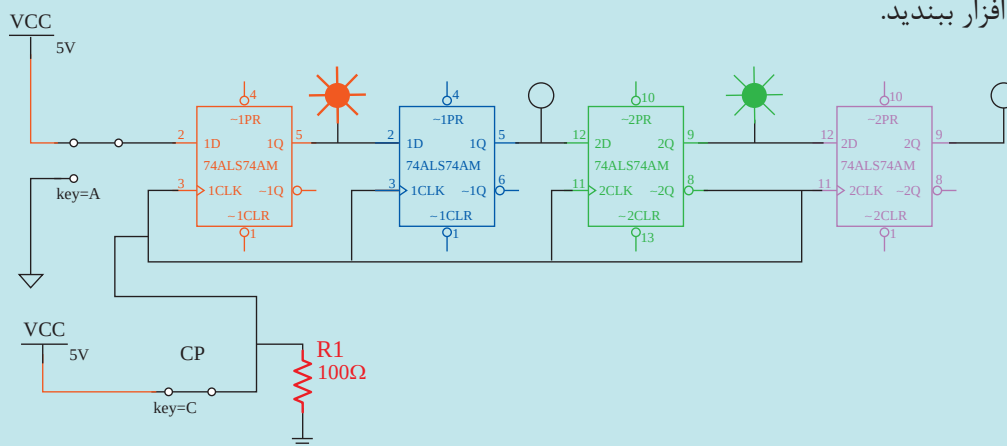


مدار شیفت رجیستر ورودی سری - خروجی موازی و شیفت رجیستر ورودی موازی - خروجی موازی در نرم افزار

هدف: بررسی عملکرد مدار شیفت رجیستر SIPO و PIPO
مواد، ابزار و تجهیزات: رایانه - نرم افزار مناسب - لوازم التحریر
مراحل اجرای کار:

۱ نرم افزار را فعال کنید.

۲ شکل ۹۶ مدار شیفت رجیستر «ورودی سری - خروجی موازی» را نشان می دهد. مدار را در محیط نرم افزار ببندید.



شکل ۹۶- مدار شیفت رجیستر ورودی سری و خروجی موازی (PISO)

جدول ۳۹- جدول صحت مدار شیفت رجیستر (PISO)

پالس ساعت	Q_1	Q_2	Q_3	Q_4
اولین				
دومین				
سومین				
چهارمین				

۳ با تغییر حالت کلیدهای A و C طبق جدول صحت ۳۹، جدول را کامل کنید.

برای نتیجه گیری دقیق از این آزمایش لازم است که پس از راه اندازی مدار، ابتدا اطلاعات اولین بیت را وارد کنید (با تغییر کلید A). سپس کلید C را ابتدا از صفر به یک سپس از یک به صفر تغییر حالت دهید (تا پالس ساعت با لبه بالا رونده عمل کند). در ادامه به همین ترتیب اطلاعات دومین بیت را وارد کنید و پالس ساعت را فعال نمایید و به همین ترتیب تا پالس چهارم این روند را ادامه دهید.

نکته

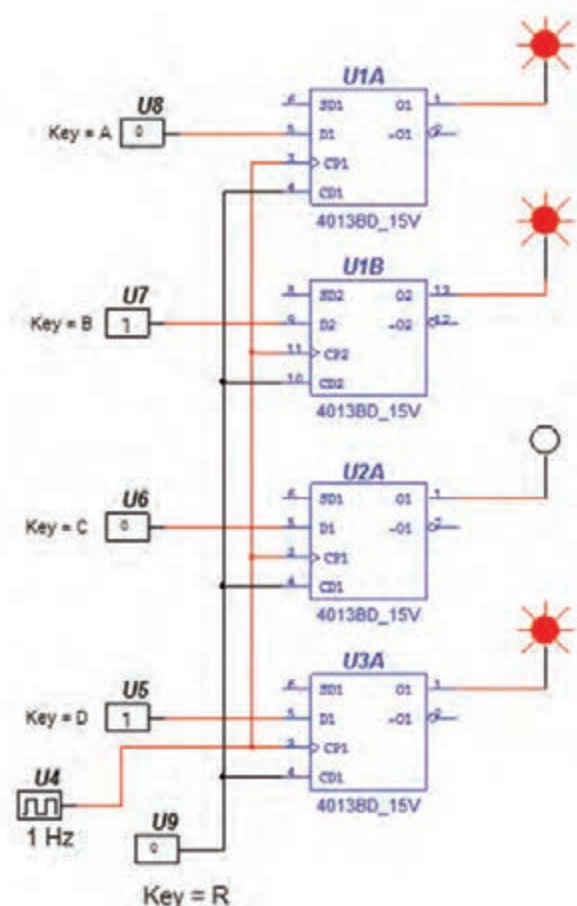


سؤال ۱: در مدار شکل ۹۶ بعد از اعمال سومین پالس چه عددی در رجیستر ذخیره می شود؟

■ شیفت رجیستر ورودی موازی - خروجی موازی

در این نوع از شیفت رجیسترها، اطلاعات ورودی هم‌زمان به خروجی انتقال می‌یابد. هنگامی که پالس ساعت صفر باشد، اطلاعات خروجی‌ها تغییری نمی‌کند. هنگامی که پالس ساعت یک باشد اطلاعات ورودی‌ها به خروجی‌ها انتقال می‌یابند.

۱ در شیفت رجیستر ورودی موازی - خروجی موازی معمولاً از فلیپ‌فلاپ نوع SR، D یا JK استفاده می‌شود. مدار شکل ۹۷ را ببینید. برای انتقال اطلاعات ورودی‌ها به خروجی‌ها ابتدا توسط Reset همه حافظه‌ها را پاک می‌کنند. این رجیستر با لبه پایین‌رونده پالس ساعت، اطلاعات ورودی‌ها را به‌طور هم‌زمان وارد حافظه‌های خروجی می‌کند.



شکل ۹۷- مدار شیفت رجیستر ورودی موازی و خروجی موازی (PIPO)

جدول ۴۰- اطلاعات ورودی مدار شیفت رجیستر (PIPO)

DATA	A	B	C	D
داده اول 1010	1	0	1	0
	Q_4	Q_3	Q_2	Q_1
داده دوم 1100	A	B	C	D
	1	1	0	0
	Q_4	Q_3	Q_2	Q_1
داده سوم 0111	A	B	C	D
	0	1	1	1
	Q_4	Q_3	Q_2	Q_1

پس از وصل کلیدها به ترتیب جدول ۴۰ اطلاعات ورودی مدار شیفت رجیستر ورودی موازی و خروجی موازی (PIPO) را کامل کنید. نتایج را که از اجرای این مدار به دست آورده‌اید، بنویسید.

.....

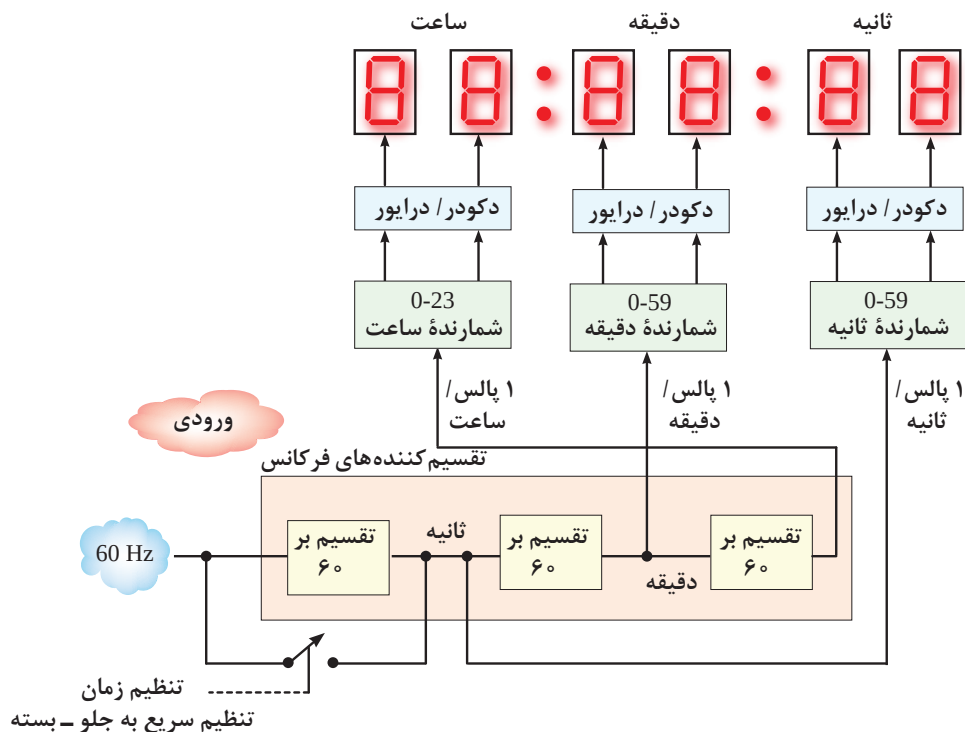
.....

.....

.....

شمارنده‌ها (Counters)

شمارنده‌ها مدارهایی هستند که از تعدادی فلیپ‌فلاپ که به صورت سری به هم متصل شده‌اند، تشکیل می‌شوند و عملاً پالس‌های ورودی به مدار را شمارش می‌کنند. عنصر اصلی شمارنده، فلیپ‌فلاپ است. شمارنده‌ها در ساعت‌های دیجیتالی، کروномتر، تایمرها در نمایشگر چراغ راهنمایی و کنتور برق دیجیتالی و موارد متعدد دیگر کاربرد دارند (شکل ۹۸).



شکل ۹۸- بلوک دیاگرام ساعت

در مورد سایر کاربردهای شمارنده تحقیق کنید و نتیجه را در قالب یک گزارش به کارگاه ارائه دهید.

تحقیق کنید



شمارنده‌ها براساس نحوه کار به شمارنده (Asynchronous) آسنکرون یا غیر هم‌زمان و شمارنده‌های (synchronous) سنکرون یا هم‌زمان تقسیم‌بندی می‌شوند. فلیپ‌فلاپ‌های به کار رفته در این نوع شمارنده از نوع T است و باید همواره $T = 1$ باشد.

در شمارنده آسنکرون، پالس ساعت فلیپ‌فلاپ‌ها به طور هم‌زمان به آنها اعمال نمی‌شود و هر طبقه پالس ساعت خود را از خروجی طبقه ما قبل خود دریافت می‌کند. در این نوع شمارنده تغییر وضعیت هر فلیپ‌فلاپ به تغییر وضعیت فلیپ‌فلاپ طبقه ماقبل بستگی دارد به همین دلیل، سرعت عمل شمارنده‌های آسنکرون نسبت به شمارنده‌های سنکرون مشابه کمتر است.

نکته مهم



برای تعیین تعداد فلیپ‌فلاپ‌ها (n) که بخواهد پایان تا عدد m شمارش کند از رابطه $2^n \geq m$ استفاده می‌کنیم. اولین عددی که در رابطه فوق صدق کند تعداد فلیپ‌فلاپ‌ها را تعیین می‌کند.

مثال: می‌خواهیم تا پایان عدد ۵۹ شمارش کنیم. تعداد فلیپ‌فلاپ‌ها را تعیین کنید.

$$2^n \leq 59$$

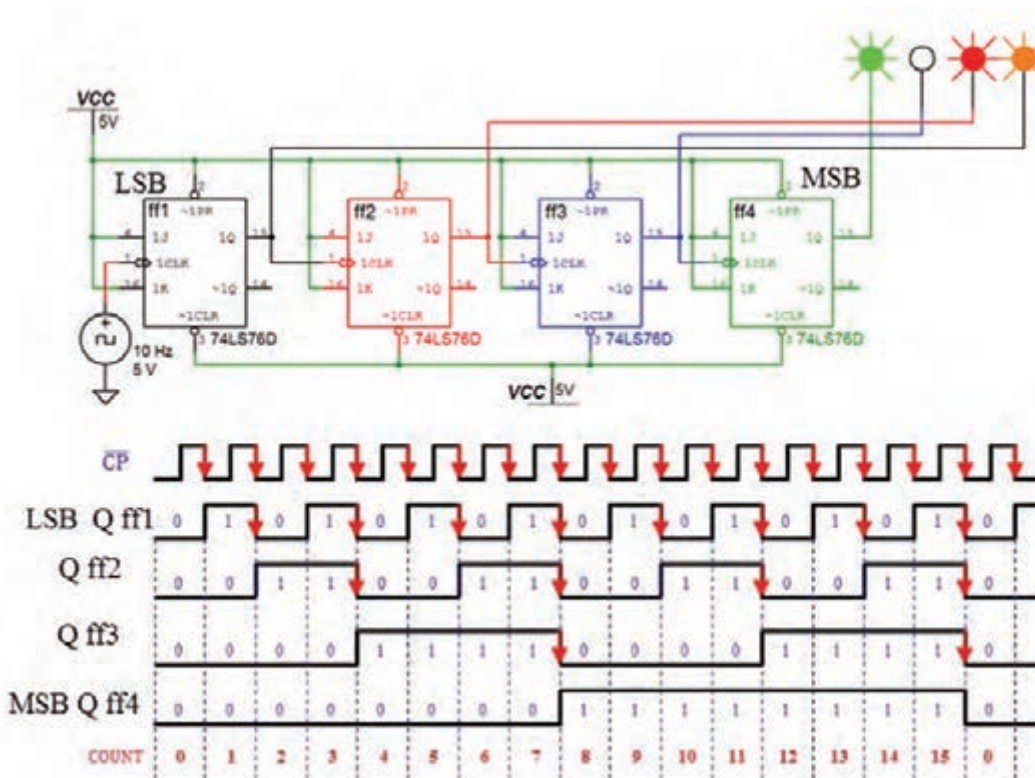
$$n = 6$$

پاسخ:

■ شمارندهٔ آسنکرون صعودی (Up Counter)

این شمارنده قادر به شمارش منظم اعداد از کم به زیاد است. به همین علت به شمارندهٔ صعودی معروف است. در شکل ۹۹ یک شمارندهٔ آسنکرون ۴ بیتی و دیاگرام زمانی نشان داده شده است که در آن از چهار فلیپ‌فلاپ نوع T استفاده شده است.

جدول ۴۱ جدول صحت این شمارنده و تغییر وضعیت خروجی‌های $Q_3Q_2Q_1Q_0$ را نشان می‌دهد. این شمارنده اعداد از صفر تا ۱۵ را شمارش می‌کند و در انتهای پالس شانزدهم تمام خروجی‌ها مساوی صفر می‌شوند و شمارنده به حالت اولیه برمی‌گردد.



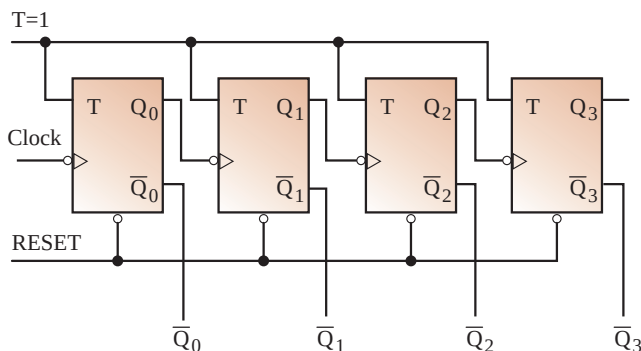
شکل ۹۹- مدار و دیاگرام زمانی شمارنده آسنکرون صعودی ۴ بیتی

جدول ۴۱- جدول صحت شمارنده آسنکرون صعودی

Clock Pluse	Q_3	Q_2	Q_1	Q_0
۰	۰	۰	۰	۰
۱	۰	۰	۰	۱
۲	۰	۰	۱	۰
۳	۰	۰	۱	۱
۴	۰	۱	۰	۰
۵	۰	۱	۰	۱
۶	۰	۱	۱	۰
۷	۰	۱	۱	۱
۸	۱	۰	۰	۰
۹	۱	۰	۰	۱
۱۰	۱	۰	۱	۰
۱۱	۱	۰	۱	۱
۱۲	۱	۱	۰	۰
۱۳	۱	۱	۰	۱
۱۴	۱	۱	۱	۰
۱۵	۱	۱	۱	۱
(Clear)	۰	۰	۰	۰
۱۶				

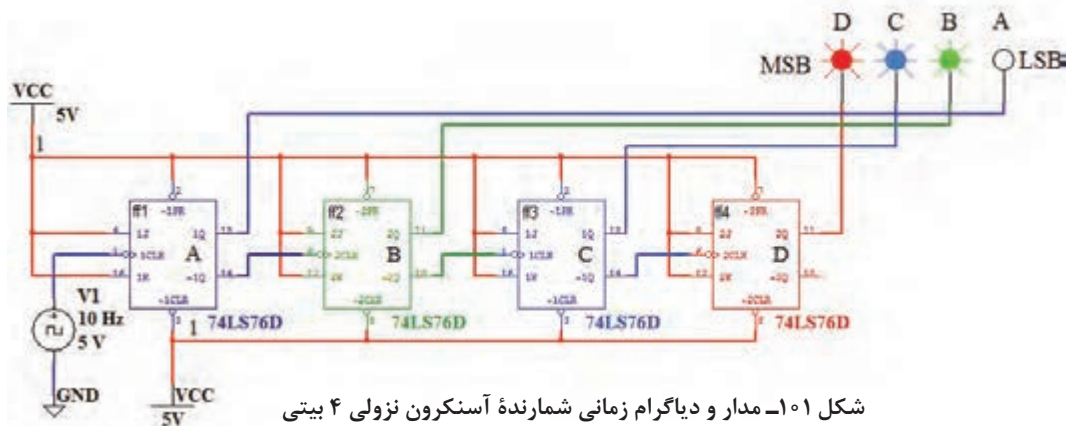
■ شمارنده آسنکرون نزولی (Down Counter)

شمارنده آسنکرون می‌تواند معکوس شمار یا نزولی باشد. یعنی از یک عدد شروع به شمارش معکوس کند و به صفر برسد. در شکل ۱۰۰ یک شمارنده آسنکرون نزولی که در آن خروجی از $\bar{Q}$ فلیپ‌فلاپ‌ها گرفته شده است را نشان می‌دهد.



شکل ۱۰۰- مدار شمارنده آسنکرون نزولی ۴ بیتی

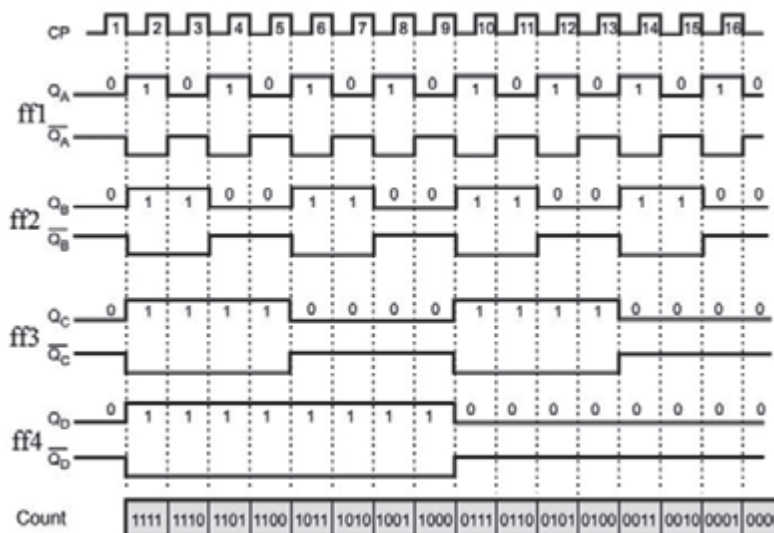
در شکل ۱۰۱ یک شمارنده آسنکرون نزولی دیگر که در آن پالس ساعت فلیپ‌فلاپ‌ها از خروجی $\bar{Q}$ فلیپ‌فلاپ قبلی تأمین می‌شود نشان داده شده است. این شمارنده از عدد ۱۵ یعنی (۱۱۱۱) شروع به شمارش می‌کند و بعد از هر پالس ساعت، یک عدد کاهش می‌یابد و در نهایت به صفر ختم می‌شود. قبل از اعمال پالس تمام خروجی‌ها در وضعیت یک قرار دارند در پایان پالس ساعت شانزدهم نیز دوباره تمام خروجی‌ها در وضعیت یک قرار می‌گیرند. در جدول ۴۲ تغییرات خروجی‌ها نشان داده شده است.



شکل ۱۰۱- مدار و دیاگرام زمانی شمارنده آسنکرون نزولی ۴ بیتی

جدول ۴۲- صحت شمارنده آسنکرون نزولی ۴ بیتی

Clock Pluse	Q_3	Q_2	Q_1	Q_0
۰	۱	۱	۱	۱
۱	۱	۱	۱	۰
۲	۱	۱	۰	۱
۳	۱	۱	۰	۰
۴	۱	۰	۱	۱
۵	۱	۰	۱	۰
۶	۱	۰	۰	۱
۷	۱	۰	۰	۰
۸	۰	۱	۱	۱
۹	۰	۱	۱	۰
۱۰	۰	۱	۰	۱
۱۱	۰	۱	۰	۰
۱۲	۰	۰	۱	۱
۱۳	۰	۰	۱	۰
۱۴	۰	۰	۰	۱
۱۵	۰	۰	۰	۰
(Reset) ۱۶	۱	۱	۱	۱



ادامه شکل ۱۰۱

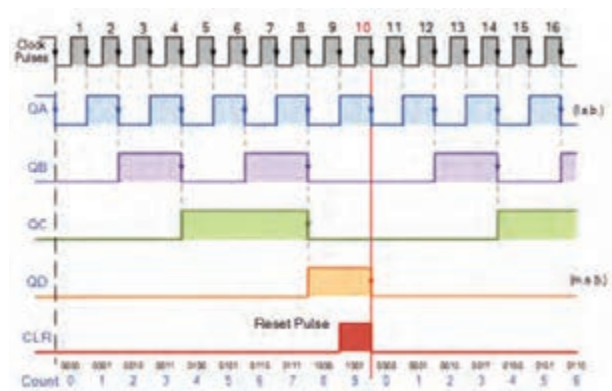
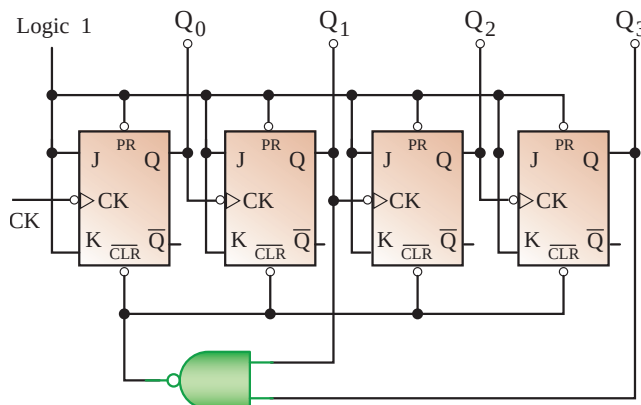
■ شمارنده آسنکرون دهدهی BCD (Binary Coded Decimal)

شمارنده آسنکرون صعودی که قبلاً بررسی شده، ۱۶ حالت مختلف دارد و می‌تواند از صفر تا عدد ۱۵ را بشمارد. در عمل نیاز به شمارنده‌هایی است که بتواند اعداد کمتر از ۱۶ حالت مثلاً از صفر تا ۱۰ را بشمارد. به این شمارنده‌ها دهدهی (اعشاری) یا BCD می‌گویند. برای طراحی یک شمارنده BCD، همان‌طور که قبلاً گفته شد ابتدا باید تعداد فلیپ‌فلاپ‌ها (N) را تعیین کنید. اگر $N=3$ بگیریم شمارنده تا عدد $2^3=8$ ، رقم می‌تواند بشمارد، برای شمارنده BCD باید $n=4$ باشد تا $2^4=16$ رقم بشمارد. بنابراین شمارنده BCD همان شمارنده آسنکرون صعودی است، با این تفاوت که باید بتواند اعداد صفر تا ۱۰ (۱۰۱۰) را بشمارد به محض رسیدن به عدد ۱۰، خروجی‌ها را پاک (Reset) کند، این عمل با گیت جانبی (کنترل) صورت می‌گیرد. در شکل ۱۰۲ مدار شمارنده آسنکرون دهدهی با استفاده از چهار فلیپ‌فلاپ T و دیاگرام زمانی آن نشان داده شده است.

در این شمارنده پس از شمارش عدد ۹ یعنی $(1001)_2$ ، خروجی فلیپ‌فلاپ‌ها به عدد $10(1010)_2$ تغییر حالت می‌دهند و هر دو ورودی گیت NAND توسط Q_1 و Q_2 برابر یک می‌شوند و خروجی آن‌را برای مدتی کوتاه به صفر می‌رسانند. به این ترتیب کلیه فلیپ‌فلاپ‌ها Reset می‌شوند و شمارش را مطابق با جدول ۴۳ از صفر شروع می‌کنند.

جدول ۴۳- صحت شمارنده
آسنکرون صعودی (ده‌دهی Mode 10)

Clock Pulse	Q_3	Q_2	Q_1	Q_0
۰	۰	۰	۰	۰
۱	۰	۰	۰	۱
۲	۰	۰	۱	۰
۳	۰	۰	۱	۱
۴	۰	۱	۰	۰
۵	۰	۱	۰	۱
۶	۰	۱	۱	۰
۷	۰	۱	۱	۱
۸	۱	۰	۰	۰
۹	۱	۰	۰	۱
(Clear)	۰	۰	۰	۰
۱۰	۰	۰	۰	۰



شکل ۱۰۲- مدار و دیاگرام زمانی شمارنده آسنکرون صعودی ۴ بیتی ده‌دهی (Mode 10)

مدار شمارنده شکل ۱۰۲ را برای شمارش حالت Mode ۱۲ اصلاح کنید.

تمرین



کار عملی ۶



مدار شمارنده‌ها در نرم‌افزار و سخت‌افزار

هدف: بررسی عملکرد مدار شمارنده آسنکرون صعودی و نزولی

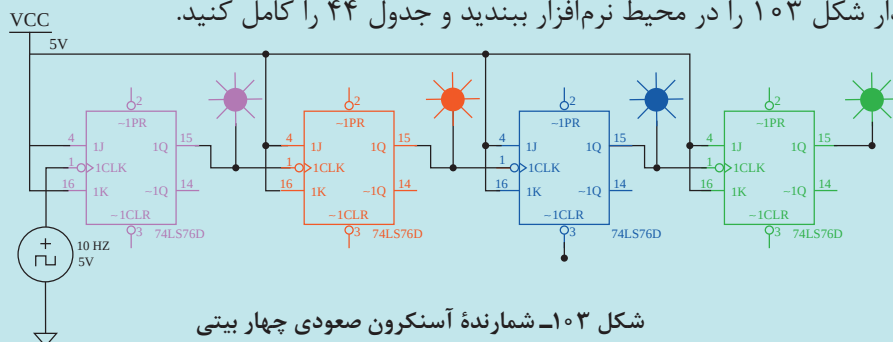
مواد، ابزار و تجهیزات: رایانه - نرم‌افزار مناسب - لوازم‌التحریر - IC۷۴۹۳ یک عدد - IC۷۴۴۷ یک عدد - نمایشگر سون‌سگمنت آند مشترک (Common Anode) - منبع تغذیه ۵ ولت DC - بردبرد و سیم رابط - مقاومت‌های ۳۳۰ اهم $\frac{1}{4}$ وات (برای هر سگمنت یا یک مقاومت 47Ω - ۱ وات سری برای پایه تغذیه نمایشگر) - مولد پالس (کلاک) یا کلید فشاری

مراحل اجرای کار شمارنده صعودی و نزولی

این شمارنده‌ها قادر به شمارش منظم اعداد از کم به زیاد و بالعکس هستند.

۱ نرم‌افزار را فعال کنید.

۲ مدار شکل ۱۰۳ را در محیط نرم‌افزار ببندید و جدول ۴۴ را کامل کنید.



شکل ۱۰۳- شمارنده آسنکرون صعودی چهار بیتی

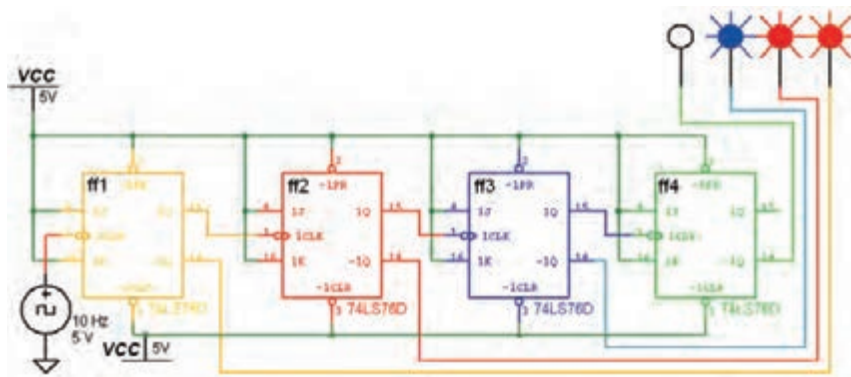
جدول ۴۴- جدول صحت مدار شمارنده صعودی چهار بیتی

عدد دسیمال	Q_4	Q_3	Q_2	Q_1	پالس ساعت
۱					
۲					
۳					
۴					
۵					
۶					
۷					
۸					
۹					
۱۰					
۱۱					
۱۲					
۱۳					
۱۴					
۱۵					
۱۶					

سؤال ۱:

۱ مدار شکل ۱۰۳ در پالس ساعت ششم و دوازدهم چه اعدادی را نشان می‌دهد؟

۲ مدار شمارنده نزولی شکل ۱۰۴ را ببینید و جدول ۴۵ صحت را کامل کنید.



شکل ۱۰۴- شمارنده نزولی چهار بیتی

جدول ۴۵- صحت مدار شمارنده نزولی چهار بیتی

عدد دسیمال	$\bar{Q}_4$	$\bar{Q}_3$	$\bar{Q}_2$	$\bar{Q}_1$	Q_4	Q_3	Q_2	Q_1	پالس ساعت
۱									
۲									
۳									
۴									
۵									
۶									
۷									
۸									
۹									
۱۰									
۱۱									
۱۲									
۱۳									
۱۴									
۱۵									
۱۶									



فرم دیگر مدار شمارنده آسنکرون نزولی را با استفاده از فلیپ‌فلاپ نوع JK ببینید و عملکرد مدار را توضیح دهید.

.....

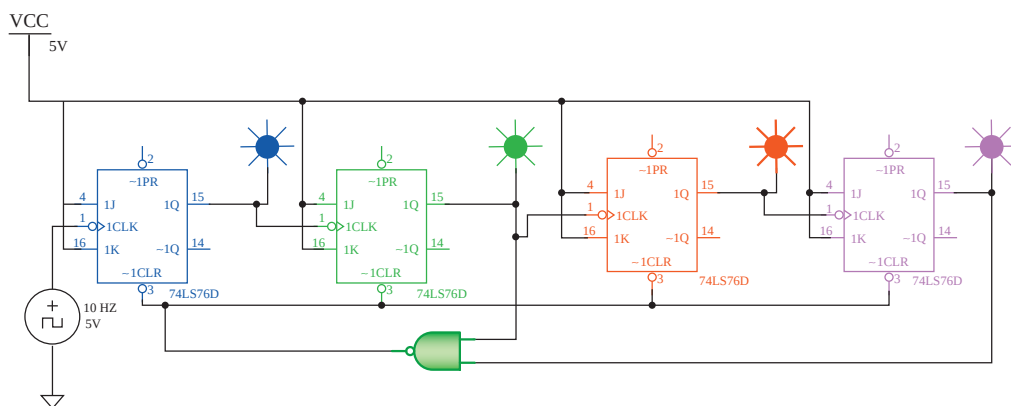
.....

.....

.....

■ **شمارنده آسنکرون ده‌دهی:** شمارنده آسنکرون ده‌دهی، همان شمارنده آسنکرون صعودی است، با این تفاوت که باید بتواند اعداد صفر تا ده را بشمارد و به محض رسیدن به عدد ده، خروجی را پاک کند. این عمل توسط یک گیت کنترل صورت می‌گیرد.

۱- مدار شکل ۱۰۵ یک شمارنده آسنکرون ده‌دهی است. مدار را ببینید و جدول ۴۶ صحت را کامل کنید.



شکل ۱۰۵- مدار شمارنده آسنکرون صعودی ۴ بیتی ده‌دهی (Mode ۱۰)

جدول ۴۶- جدول صحت مدار شمارنده آسنکرون ده‌دهی

Q ₃	Q ₂	Q ₁	Q ₀	Decimal
۰	۰	۰	۰	
۰	۰	۰	۱	
۰	۰	۱	۰	
۰	۰	۱	۱	
۰	۱	۰	۰	
۰	۱	۰	۱	
۰	۱	۱	۰	
۰	۱	۱	۱	
۱	۰	۰	۰	
۱	۰	۰	۱	

■ اجرای سخت‌افزاری مدار شمارنده با IC های ۷۴۲۹۳ و ۷۴۴۷ همراه با سون سگمنت: IC۷۴۲۹۳

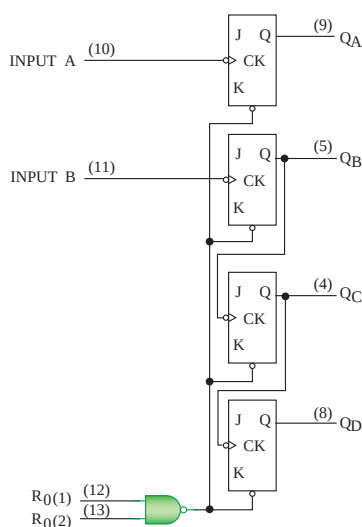
یک شمارنده دودویی ۴ بیتی است. این IC از ۴ فلیپ‌فلاپ نوع JK تشکیل شده است. ولتاژ تغذیه این IC از ۴.۷۵ تا ۵.۲۵ ولت می‌باشد. با توجه به شکل ۱۰۶ برگه اطلاعات و مدار داخلی IC۷۴۲۹۳ این شمارنده از نوع آسنکرون (Ripple Counter) است؛ یعنی تغییر وضعیت هر فلیپ‌فلاپ وابسته به خروجی فلیپ‌فلاپ قبلی است، از این IC معمولاً برای ساخت یک شمارنده دودویی ۴ بیتی استفاده می‌کنند. در این حالت، مدار می‌تواند اعداد را به صورت دودویی (۰۰۰۰ تا ۱۱۱۱) یعنی از (۰ تا آخر ۱۵) بشمارد؛ به ترتیب مقادیر ۰۰۰۰، ۰۰۰۱، ۰۰۱، ... تا ۱۱۱۱ تغییر می‌کند. یکی از ویژگی‌های مهم این آی‌سی، قابلیت کار در مدهای مختلف است:

- در حالت مد ۸، شمارنده فقط از ۰ تا ۷ می‌شمارد. این کار با تنظیم نحوه اعمال کلاک انجام می‌شود.
 - در حالت مد ۲، خروجی بین ۰ و ۱ تغییر می‌کند؛ یعنی فرکانس ورودی را به ۲ تقسیم می‌کند.
- به همین دلیل، IC۷۴۲۹۳ در کاربردهایی مانند تقسیم فرکانس، مدارهای زمان‌سنجی و سیستم‌های شمارش دیجیتال مورد استفاده می‌شود.

'293, 'LS293
RESET/COUNT FUNCTION TABLE

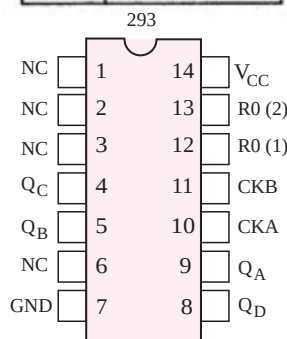
RESET INPUTS		OUTPUT			
R ₀ (1)	R ₀ (2)	Q _D	Q _C	Q _B	Q _A
H	H	L	L	L	L
L	X	COUNT			
X	L	COUNT			

NOTES: A. Output Q_A is connected to input B for BCD count.
B. Output Q_D is connected to input A for bi-quinary count.
C. Output Q_A is connected to input B.
D. H = high level, L = low level, X = irrelevant



'293, 'LS293
COUNT SEQUENCE
(See Note C)

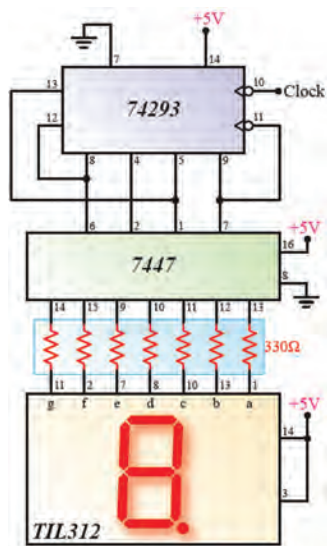
COUNT	OUTPUT			
	Q _D	Q _C	Q _B	Q _A
0	L	L	L	L
1	L	L	L	H
2	L	L	H	L
3	L	L	H	H
4	L	H	L	L
5	L	H	L	H
6	L	H	H	L
7	L	H	H	H
8	H	L	L	L
9	H	L	L	H
10	H	L	H	L
11	H	L	H	H
12	H	H	L	L
13	H	H	L	H
14	H	H	H	L
15	H	H	H	H



شکل ۱۰۶- برگه اطلاعات IC۷۴۲۹۳



با IC ۷۴۲۹۳ چند Mode قابل شمارش می‌توان طراحی کرد؟



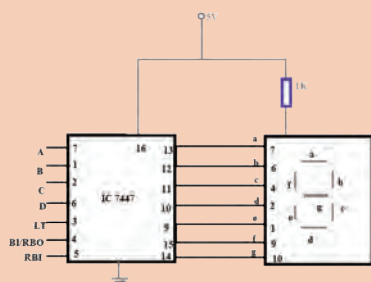
شکل ۱۰۷- مدار شمارنده

۱- مدار شکل ۱۰۷ را که یک شمارنده است، روی برد برد ببندید.

۲- ترتیب اعداد شمارش شده را در یک جدول بنویسید و در مورد عملکرد آن توضیح دهید.



چنانچه برای صرفه‌جویی در هزینه و زمان اجرای مدار سخت‌افزاری شمارنده به جای مقاومت‌های ۳۳۰ اهمی هر قطعه دیود هفت قطعه‌ای از یک مقاومت ۱kΩ در خط تغذیه مطابق شکل ۱۰۸ استفاده شود، اشکالی در مدار حاصل می‌شود، آن را شرح دهید.



شکل ۱۰۸- مدار راه‌انداز (رمزگشا) هفت قطعه‌ای

درباره نحوه طراحی و ساخت شمارنده ۰ تا ۹۹ با استفاده از تراشه SN۷۴۲۹۳ و مراحل پیاده‌سازی آن به‌طور کامل بررسی لازم را انجام دهید، سپس گزارش آن را به هنرآموز کارگاه ارائه کنید.

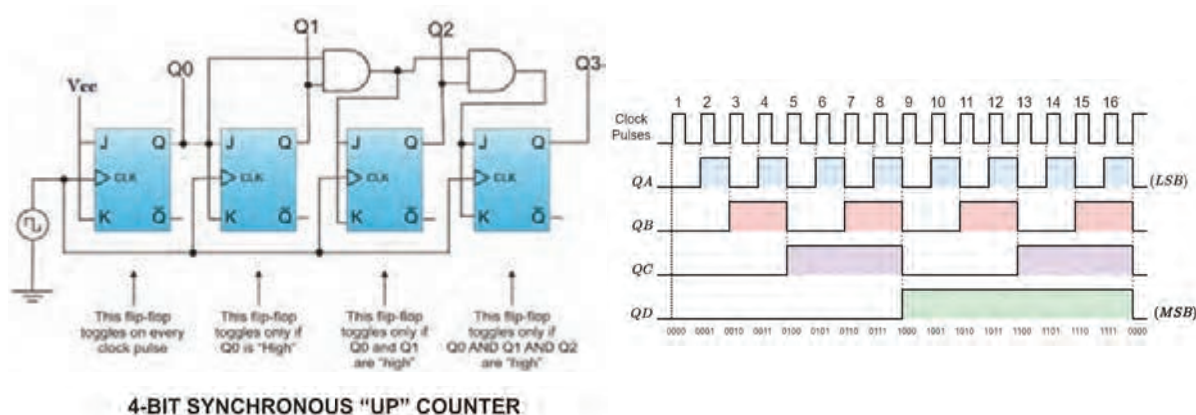


■ مدار شمارنده سنکرون (Synchronous) چهار بیتی

مدار شمارنده سنکرون چهار بیتی می‌تواند اعداد دودویی را از ۰ تا ۱۵ به‌صورت منظم شمارش کند. در این مدار معمولاً از چهار فلیپ‌فلاپ استفاده می‌شود که همگی به یک سیگنال پالس ساعت مشترک متصل هستند. سنکرون بودن مدار به این معناست که همه فلیپ‌فلاپ‌ها به‌طور هم‌زمان و با لبه فعال کلاک تغییر وضعیت می‌دهند. این ویژگی باعث افزایش دقت عملکرد نسبت به شمارنده‌های آسنکرون می‌شود.

شمارنده چهار بیتی در سیستم‌های دیجیتال برای تقسیم فرکانس، تولید توالی پالس و زمان‌بندی کاربرد گسترده‌ای دارد. طراحی این مدار معمولاً با استفاده از فلیپ‌فلاپ‌های D, JK یا T و به کمک نقشه کارنو برای ساده‌سازی معادلات منطقی انجام می‌شود.

در مدار شکل ۱۰۹ فلیپ‌فلاپ اول (کم‌ارزش‌ترین بیت) با هر پالس ساعت تغییر وضعیت می‌دهد. خروجی فلیپ‌فلاپ دوم فقط زمانی تغییر می‌کند که خروجی فلیپ‌فلاپ اول از ۰ به ۱ برگردد، یعنی هنگام شمارش عدد زوج. به همین ترتیب، فلیپ‌فلاپ سوم تغییر می‌کند هنگامی که دو بیت کم‌ارزش‌تر هر دو از ۰ به ۱ برگردند. در نهایت، فلیپ‌فلاپ چهارم (با ارزش‌ترین بیت MSB) زمانی تغییر وضعیت می‌دهد که سه بیت پایین‌تر همگی از ۰ به ۱ تغییر کنند، و شمارنده از مقدار ۰۰۰۰ به ۱۱۱۱ برسد.



شکل ۱۰۹- مدار و دیاگرام شمارنده سنکرون چهار بیتی

با توجه به دیاگرام زمانی توضیح دهید چرا خروجی‌های Q_A, Q_B, Q_C, Q_D تمام فلیپ‌فلاپ‌ها به صورت هم‌زمان (سنکرون) تغییر می‌کنند، ولی تغییر مقدارشان به شرط خروجی‌های فلیپ‌فلاپ‌های قبلی وابسته است؟

پرسش

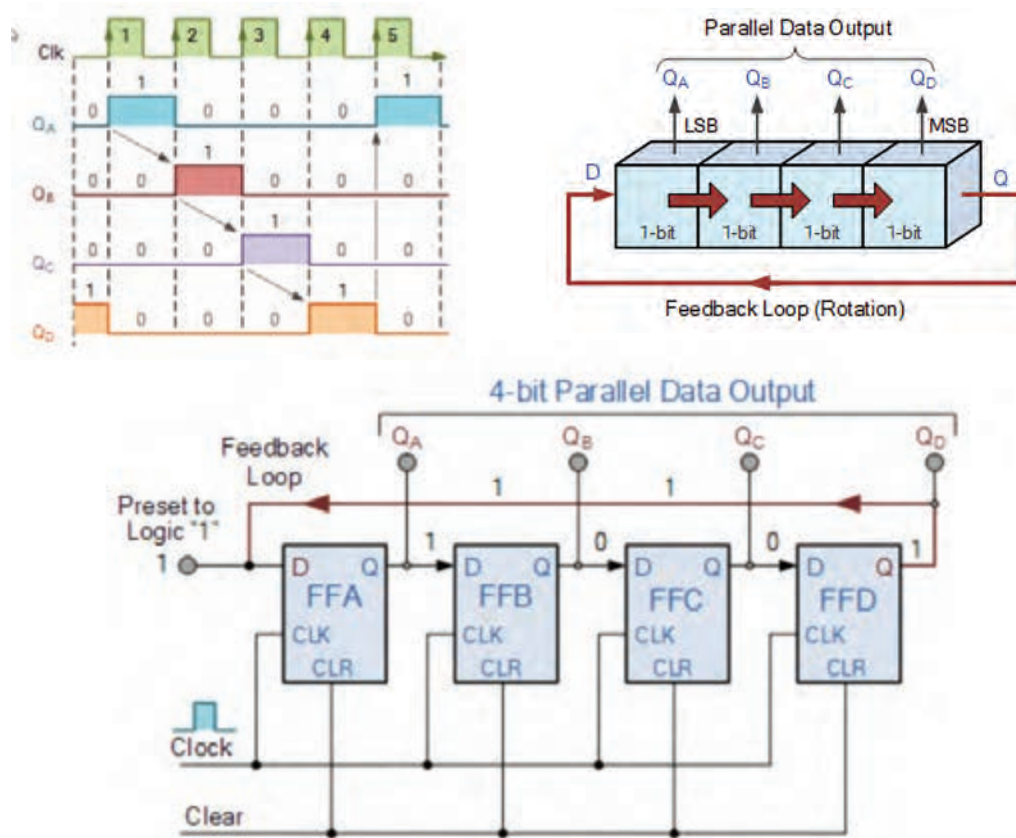


■ شمارنده حلقوی (Ring counter)

شمارنده حلقوی شکل ۱۱۰ از ترکیب فلیپ‌فلاپ‌های نوع D به گونه‌ای شکل می‌گیرد که خروجی آخرین فلیپ‌فلاپ به ورودی اولین فلیپ‌فلاپ فیدبک شده است. این شمارنده از نوع سنکرون است؛ ابتدا تمام فلیپ‌فلاپ‌ها را به وسیله خط Clear (پاک) Reset می‌کنیم.

سپس، پیش از اعمال پالس‌های کلاک، یک پالس PRESET به ورودی اولین فلیپ‌فلاپ (FFA) اعمال می‌گردد. به این ترتیب، یک مقدار منطقی «۱» در مدار شمارنده حلقوی (Load) قرار داده می‌شود. در ادامه، با هر پالس ساعت بعدی، شمارنده همان بیت داده را بین چهار فلیپ‌فلاپ به گردش درمی‌آورد و این چرخش هر چهار سیکل پالس ساعت یک بار در اطراف «حلقه» تکرار می‌شود.

به این نوع جابه‌جایی داده (چرخش Rotation) گفته می‌شود. مشابه رجیستر انتقال داده (Shift Register) که قبلاً شرح داده شد، حرکت بیت داده از چپ به راست در یک شمارنده حلقوی را می‌توان به صورت گرافیکی، همراه با نمودار زمانی آن، در شکل ۱۱۰ مشاهده کنید.

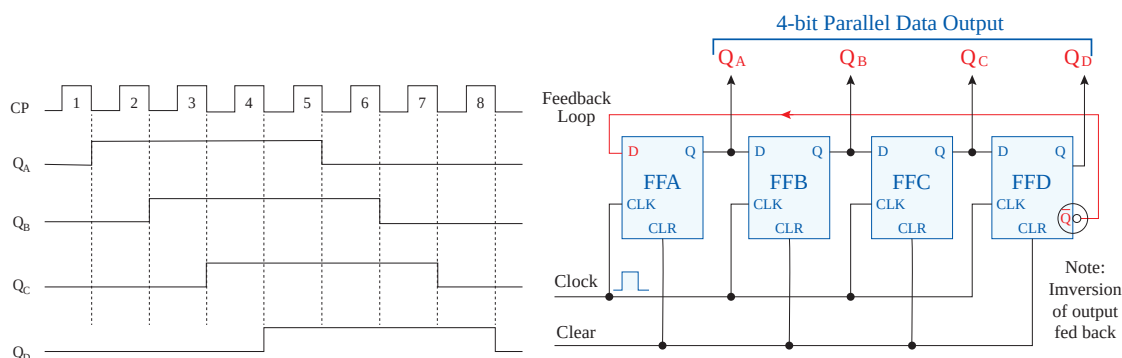


شکل ۱۱۰- مدار و دیاگرام شمارنده سنکرون چهاربیتی حلقوی

■ شمارنده جانسون (Johnson counter)

در شکل ۱۱۱ شمارنده جانسون ۴ بیتی را مشاهده می‌کنید در این مدار شمارنده از فلیپ‌فلاپ نوع D استفاده شده است. این شمارنده یک شیفت رجیستر با ورودی سری و خروجی سری است. که در آن $\bar{Q}$ آخرین فلیپ‌فلاپ به ورودی اولین فلیپ‌فلاپ متصل شده است. شمارنده جانسون چهار بیتی دارای ۸ حالت مختلف است و شمارنده جانسون ۵ بیتی دارای ۱۰ حالت مختلف است پس شمارنده جانسون n بیتی دارای $2n$ حالت است. (n تعداد طبقات شمارنده را مشخص می‌کند)

اتصال فیدبک از $\bar{Q}$ فلیپ‌فلاپ چهارم به ورودی D فلیپ‌فلاپ اول باعث می‌شود شمارنده به شیوه‌ای متفاوت «شمارش» کند. برخلاف شمارنده حلقوی معمولی که به صورت: ۱ و ۲ و ۴ و ۸ و ۱۰۰۰، ۰۱۰۰، ۰۰۱۰، ۰۰۰۱ شمارش می‌کند.



شکل ۱۱۱- مدار و دیاگرام

جدول ۴۷- حالات مختلف شمارندهٔ جانشون چهاربیتی

Clock Pluse No	Q _A	Q _B	Q _C	Q _D	حالات مختلف شمارش
۰	۰	۰	۰	۰	صفر ۰۰۰۰
۱	۱	۰	۰	۰	هشت ۱۰۰۰
۲	۱	۱	۰	۰	دوازده ۱۱۰۰
۳	۱	۱	۱	۰	چهارده ۱۱۱۰
۴	۱	۱	۱	۱	پانزده ۱۱۱۱
۵	۰	۱	۱	۱	هفت ۰۱۱۱
۶	۰	۰	۱	۱	سه ۰۰۱۱
۷	۰	۰	۰	۱	یک ۰۰۰۱
۸	۰	۰	۰	۰	صفر ۰۰۰۰

در شمارنده جانشون، عمل شمارش به صورت افزایشی و سپس کاهش انجام می شود؛ در جدول ۴۷ چگونگی شمارش، شمارنده جانشون را مشاهده می کنید.

بستن مدارهای شمارنده های سنکرون به صورت نرم افزاری

هدف: کسب مهارت در شناخت عملکرد مدارهای شمارنده های سنکرون حلقوی و جانشون

مواد، ابزار و تجهیزات: لوازم التحریر - اینترنت - رایانه - نرم افزار مولتی سیم یا هر نرم افزار مناسب دیگر
مراحل اجرای کار:

■ شمارنده حلقوی از ترکیب فلیپ فلاپ های نوع D به گونه ای شکل می گیرد که خروجی آخرین فلیپ فلاپ به ورودی اولین فلیپ فلاپ فیدبک شده است.

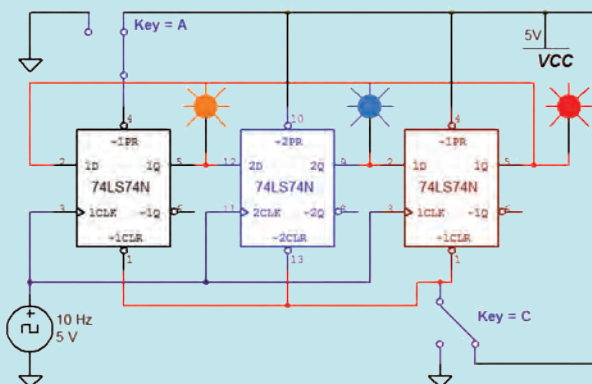
1 مدار شکل ۱۱۲ شمارنده حلقوی سه بیتی را نشان می دهد. این مدار را ببینید و جدول ۴۸ صحت را کامل کنید

کار عملی



جدول ۴۸- صحت شمارنده حلقوی سه‌بیتی

شمارش دهدهی خروجی	Q_2	Q_1	Q_0	پالس ساعت
۰				
۱				
۲				
۳				
۴				



شکل ۱۱۲- شمارنده سنکرون حلقوی سه‌بیتی

با استفاده از چهار فلیپ‌فلاپ نوع D یک شمارنده حلقوی چهار بیتی را ببندید، مراحل کار را توضیح دهید و جدول صحت آن را کامل نمایید.

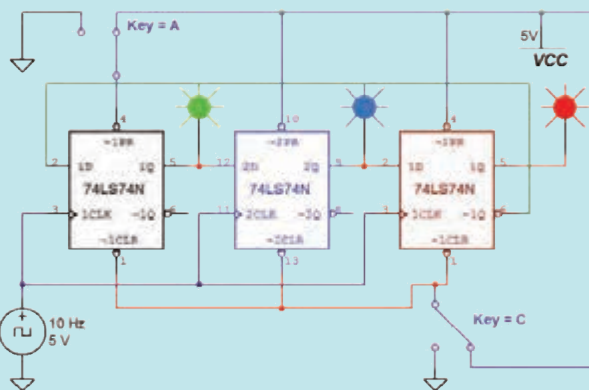
فعالیت



۲ مدار شمارنده جانسون شکل ۱۱۳ را ببندید. در این مدار خروجی معکوس آخرین $\bar{Q}$ فلیپ‌فلاپ به ورودی اولین فلیپ‌فلاپ وصل شده است. در این شمارنده از فلیپ‌فلاپ نوع D استفاده شده است. این شمارنده دارای شش حالت مختلف است. پس از بستن مدار، جدول ۴۹ صحت را کامل کنید.

جدول ۴۹- صحت شمارنده حلقوی سه‌بیتی

شمارش دهدهی خروجی	Q_2	Q_1	Q_0	پالس ساعت
۰				
۱				
۲				
۳				
۴				



شکل ۱۱۳- شمارنده جانسون سه‌بیتی

مدار شمارنده جانسون پنج بیتی را ببندید و جدول صحت آن را کامل نمایید و گزارش آن را به کارگاه ارائه دهید.

فعالیت





به نظر شما در دستگاه کپی برای کنترل شماره گان کپی اوراق از چه مدارهای ترکیبی و ترتیبی استفاده می شود؟ از طریق بارش فکری موضوع را بررسی و جمع بندی کنید.

آبیاری نوبتی چهار بخش گلخانه با استفاده از شمارنده حلقوی

شرح اولیه پروژه



شکل ۱۱۴- آبیاری گلخانه های آموزشی یا صنعتی فضا

در گلخانه های آموزشی یا صنعتی، معمولاً فضا به چند بخش تقسیم می شود تا انواع مختلف گیاهان در شرایط مناسب نگهداری شوند (شکل ۱۱۴).

اگر تمام بخش ها به طور هم زمان آبیاری شوند:

■ فشار آب کاهش می یابد.

■ مصرف انرژی افزایش می یابد.

■ کنترل زمان آبیاری دشوار می شود.

برای حل این مشکل، می توان از یک سیستم آبیاری نوبتی استفاده کرد؛ به این صورت که در هر لحظه فقط یک بخش گلخانه آبیاری شود و پس از پایان زمان تعیین شده، آبیاری به بخش بعدی منتقل شود. در این پروژه چهار باغچه نیاز به آبیاری خودکار دارد، بنابراین از شمارنده حلقوی ۴ بیتی استفاده می کنیم تا چهار شیر برقی را به صورت چرخشی و منظم کنترل کنیم (شکل ۱۱۵).



شکل ۱۱۵- آبیاری خودکار گلخانه با چهار فضای (باغچه) مجزا

اجرای عملی

هدف: راه اندازی مدار آبیاری ترتیبی چهار بخش با استفاده از شمارنده حلقوی.

مواد، ابزار و تجهیزات: رایانه - نرم افزار مناسب - لوازم التحریر

■ منبع تغذیه ۵ یا ۱۲ ولت

■ IC شمارنده حلقوی یا شیفتر رجیستر (Dual D Flip-Flop-IC۷۴LS۷۴) دو عدد

■ ۴ عدد LED به جای شیر برقی برای تست اولیه

■ ۴ عدد مقاومت مناسب LED و مابقی مطابق نقشه پروژه

■ برد بُرد

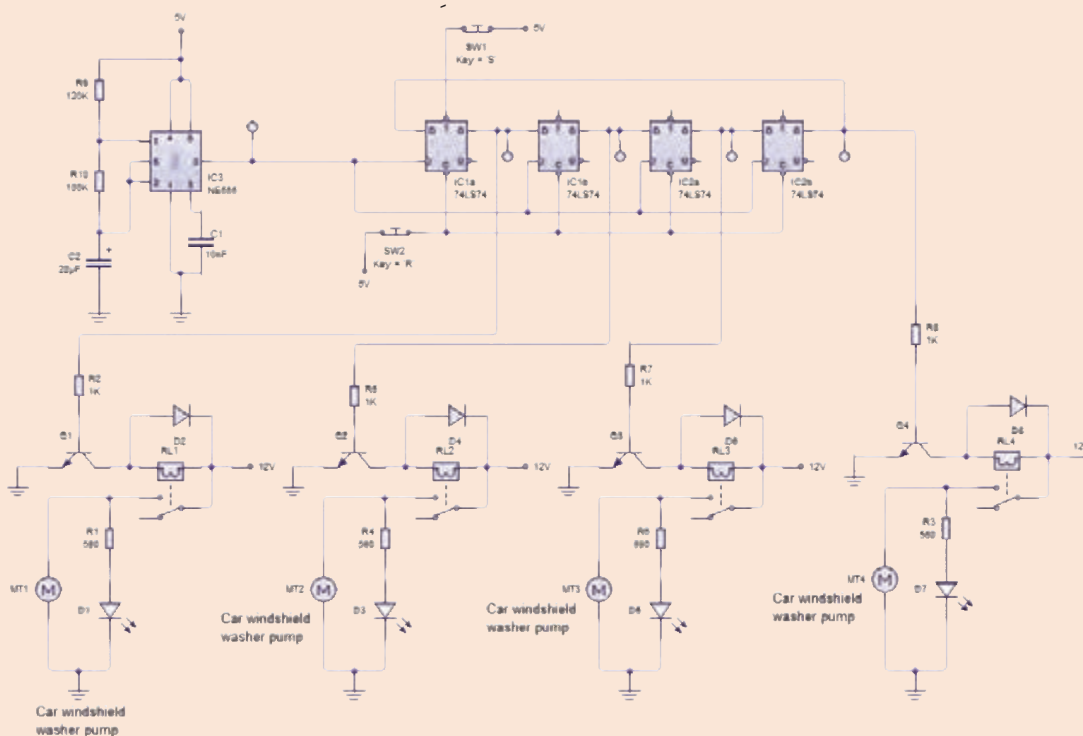
■ سیم رابط

■ مولد پالس (IC ۵۵۵)

■ ۴ رله یا درایور ترانزیستوری + شیر برقی (آب پاش و پمپ آب شیشه شور خودرو)

مراحل اجرای کار:

- ۱ بررسی تئوری مدار
 - مفهوم شمارنده حلقوی را مرور کنید.
 - تحلیل پروژه، چرا در هر لحظه فقط یک خروجی فعال است.
- ۲ بستن مدار شمارنده
 - نقشه مدار الکترونیکی شکل در نرم‌افزار مناسب ببینید و با شبیه‌سازی آن از صحت عملکرد آن اطمینان حاصل شود.
 - هر رله نماینده یک بخش گلخانه باشد.
- ۳ فهرست قطعات موردنیاز را تهیه کنید.
- ۴ مدار الکترونیکی پروژه انتخاب شده شکل ۱۱۶ را بر روی برد ببینید و آن را آزمایش کنید.



شکل ۱۱۶- نقشه مدار پروژه

- ۵ نقشه الکترونیکی مدار را به محیط نرم‌افزار آلیتوم دیزاینر انتقال دهید سپس با رعایت استانداردها و نکات طراحی، نقشه PCB آن را رسم کنید.
- ۶ فیبر مدار چاپی (اندازه فیبر مدار چاپی با پیشنهاد هنرآموز کارگاه تعیین شود) تهیه شود.
- ۷ تمامی قطعات بر روی برد نصب و لحیم کاری پایه‌های آنها صورت گیرد.
- ۸ با تنظیم مقدار ولتاژ تغذیه مدار، آن را به ورودی مدار اتصال دهید و خروجی مدار منطقی را با آزمایش مشاهده کنید.
- ۹ نتایج مورد انتظار از اجرای پروژه را بررسی کنید و در گزارش کار خود بیاورید.
- سیستم به صورت ترتیبی چهار بخش را فعال می‌کند.
- در هر لحظه فقط یک بخش آبیاری می‌شود.

- مدار ساده و قابل توسعه است.
- چنانچه بخواهیم ۶ بخش آبیاری داشته باشیم چه تغییری در مدار باید ایجاد شود.
- اگر بخواهیم زمان آبیاری هر بخش متفاوت باشد چه راهکاری پیشنهاد شود.
- ۱۰ گزارش کار انجام پروژه را به صورت فایل رایانه‌ای مستندسازی کنید، آن را برای ارزشیابی به هنرآموز خود ارائه دهید.

خودارزیابی

- ۱ مدارهایی که حافظه دارند را می‌گویند.
- ۲ مدارهای ترتیبی اطلاعات جدید را چگونه پذیرش می‌کنند؟ توضیح دهید.
- ۳ انواع فلیپ‌فلاپ و کاربرد آنها را تشریح کنید.
- ۴ SIPO چه نوع شیفت رجیستری است؟ نام ببرید.
- ۵ کدام آی‌سی، شمارنده است؟

۷۴۷۴-۱ ۷۴۲۹۳-۲ ۷۴۱۵۱-۳ ۷۴۸۳-۴

- ۶ جدول درستی مربوط به کدام فلیپ‌فلاپ است؟

S-R-۱ J-K-۲ J-K MS-۳ D-۴

- ۷ جدول درستی فلیپ‌فلاپ T کدام گزینه است؟

	CP	$Q_{(t)}$
0		$Q_{(t-1)}$
1		$\bar{Q}_{(t-1)}$

-۴

	CP	Q
	0	0
	1	1

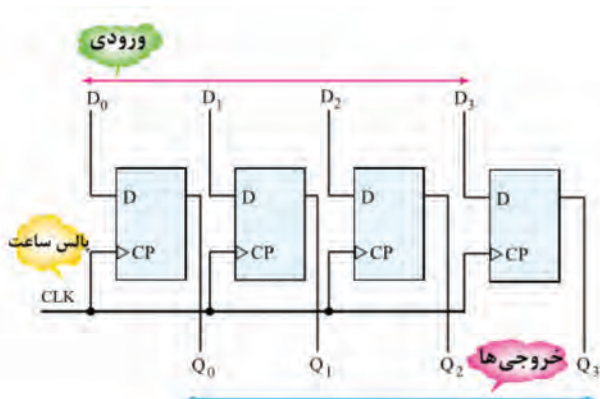
-۳

		$Q_{(t)}$
0	0	$Q_{(t-1)}$
0	1	0
1	0	1
1	1	$\bar{Q}_{(t-1)}$

-۲

Q		
0	0	تغییر نمی‌کند
0	1	Reset
1	0	Set
1	1	غیرمجاز

-۱

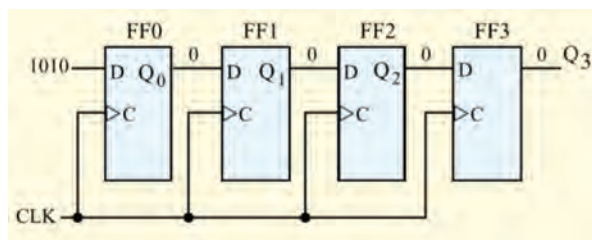


- ۸ نوع شیفت رجیستر کدام گزینه است؟

SIPO-۲ SISO-۱
PISO-۴ PIPO-۳

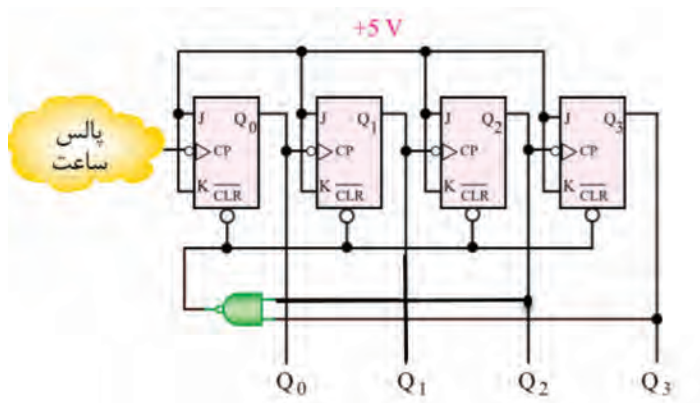
۹. نوع شیفت رجیستر کدام گزینه است؟

- ۱- PISO ۲- SIPO
۳- SISO ۴- PIPO



۱۰. شمارنده تا پایان چه عددی شمارش می‌کند؟

- ۱- ۱۰ ۲- ۱۱
۳- ۱۲ ۴- ۱۳



۱۱. برای شمارش تا پایان عدد ۸۵ به چند فلیپ فلاپ نیاز داریم؟

- ۱- ۴ ۲- ۵ ۳- ۶ ۴- ۷

ارزشیابی شایستگی پیاده سازی مدارهای ترکیبی

نام و نام خانوادگی	شماره ملی	تاریخ ارزشیابی:	نوبت:
کد حرفه	۸۲۱۲۱۲۰۳۱	استاندارد عملکرد: پیاده سازی مدارهای ترکیبی شامل انواع فلیپ فلاپ (T ⁺ D ⁺ JK ⁺ S) ثبات ها، شمارنده و انواع آن، با استفاده از محیط نرم افزار و برد بژد و ساخت پروژه مدار دیجیتال.	۲
کد وظیفه شغل	۸۲۱۲۱۲۰۳۰۲	مهارت	۲
کد کار	۸۲۱۲۱۲۰۳۳۲	مهارت	۲

ردیف	مراحل کار	شرایط انجام کار (ابزار، مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص ها/دآوری /نمره دهی)	نتایج ممکن	نمره کسب شده
۱	تشریح انواع فلیپ فلاپ ها، ثبات ها و شمارنده ها با در نظر گرفتن انواع مختلف آنها براساس نیازمندی های پروژه	نمودارهای زمانی، جدول مشخصه فلیپ فلاپ ها، برگه و زمان ۳۰-۲۵ دقیقه در کلاس درس یا کارگاه	آزمون، بررسی ترسیم جدول و نمودار زمانی، پرسش کاربرد انواع فلیپ فلاپ، چک لیست	تشریح دقیق و کاربرد صحیح در پروژه	۳	
۲	شبیه سازی و تحلیل عملکرد مدارهای ترکیبی طراحی شده در محیط نرم افزاری	رایانه، نرم افزار شبیه سازی مدارهای دیجیتال، فایل یا طرح مدار ترکیبی، دفترچه یادداشت، کتاب، زمان ۳۰-۲۵ دقیقه در کارگاه الکترونیک مجهز به رایانه	اجرای صحیح شبیه سازی، تطابق عملکرد مدار با طراحی و تشخیص خطاهای احتمالی	شبیه سازی و تحلیل صحیح عملکرد مدار	۳	
۳	پیاده سازی مدارهای ترکیبی (فلیپ فلاپ ها، ثبات ها، شمارنده ها) بر روی بردبژد با استفاده از قطعات الکترونیکی و گیت های منطقی	بردبورد، آی سی فلیپ فلاپ و گیت منطقی، منبع تغذیه، سیم های رابط، LED و مقاومت، طرح مدار و زمان حدود ۴۰-۳۰ دقیقه در کارگاه الکترونیک	بررسی عملکرد مدار، خروجی ها و تغییر حالت ها، چک لیست اتصالات و پرسش شفاهی	پیاده سازی کامل و عملکرد صحیح مدار	۳	
۴	طراحی و ساخت یک پروژه مدار دیجیتال شامل مدارهای ترکیبی با رعایت اصول فنی و استانداردهای مربوطه، مستندسازی و گزارش دهی فنی	رایانه، بردبورد، آی سی منطقی و فلیپ فلاپ، منبع تغذیه، سیم رابط، LED و مقاومت، ابزار اندازه گیری و زمان حدود ۴۵-۳۰ دقیقه در کارگاه الکترونیک	بررسی پروژه و مدار ساخته شده، چک لیست طراحی و پیاده سازی، کنترل گزارش فنی و مستندات، ارائه شفاهی پروژه	طراحی، ساخت و مستندسازی کامل	۳	
	شایستگی های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:	آزمایشگاه مدارهای منطقی یا سایت رایانه، نرم افزار شبیه سازی مدار، تجهیزات کارگاهی و حفاظت فردی و دستور العمل های ایمنی و بهداشت حرفه ای	مشاهده رفتار هنجار در حین انجام فعالیت، میزان رعایت اصول ایمنی و نظم، فعالیت های گروهی، مدیریت زمان و مسئولیت پذیری	رعایت کامل ایمنی، بهداشت و محیط زیست، رفتار حرفه ای، نظم و مسئولیت پذیری بالا	۲	
				رعایت نسبی ایمنی و بهداشت با خطاهای جزئی، رفتار کاری مناسب	۱	
ارزشیابی کار (شایستگی انجام کار)						
☐ بلی ☐ خیر						

توضیحات:

۱- معیار شایستگی انجام کار:

- کسب حداقل نمره ۲ در تمامی مراحل بحرانی شامل تشریح مدارهای ترکیبی، شبیه سازی و تحلیل مدارهای ترکیبی، پیاده سازی عملی مدارها، و طراحی و ساخت پروژه مدار دیجیتال است. (مراحل بحرانی).

- کسب حداقل نمره ۲ از بخش شایستگی های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش

- کسب حداقل میانگین ۲ از مراحل کار

۲- تعریف سطوح شایستگی: صرف نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه ای انجام می شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

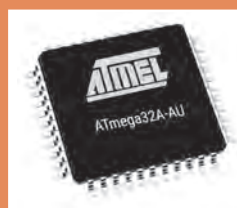
مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه ریزی و تحلیل، پاسخ گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب یابی، هدایت و راهنمایی دیگران.



پودمان چهارم

انتخاب میکروکنترلر (Microcontroller) و برنامه‌ریزی آن



میکروکنترلر یک مدار مجتمع (IC) است که درون یک تراشه کوچک، بخش‌های اصلی یک سیستم رایانه‌ای را در خود جای می‌دهد. به دلیل دارا بودن پردازنده مرکزی (CPU)، حافظه برنامه (معمولاً Flash یا ROM) برای ذخیره‌کد، حافظه داده (RAM) برای پردازش موقت و ورودی/خروجی‌های قابل برنامه‌ریزی. میکروکنترلرها را گاهی رایانه‌های تک‌تراشه‌ای (Single-Chip Microcomputers) می‌نامند.

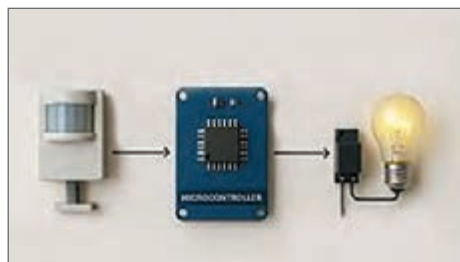
این تراشه‌ها می‌توانند داده‌ها را از محیط (مثلاً از طریق سنسورها) دریافت کرده، آنها را پردازش نمایند و در نهایت خروجی مناسبی (مانند روشن کردن یک موتور یا ارسال داده) تولید کنند. با استفاده از میکروکنترلرها، بسیاری از مدارهای کنترلی ساده تا متوسط که پیش از این نیازمند ده‌ها قطعه دیجیتال و آنالوگ بودند امروزه با یک تراشه مرکزی و چند عنصر کمکی قابل پیاده‌سازی‌اند.

میکروکنترلرها در کاربردهای گوناگونی مانند سیستم‌های کنترلی صنعتی، لوازم خانگی هوشمند (مثل یخچال یا ماشین لباسشویی)، تجهیزات پزشکی، رباتیک، اینترنت اشیا (IoT)، سیستم‌های آبیاری هوشمند در کشاورزی و سایر سیستم‌های الکترونیکی که در تجهیزات و دستگاه‌های مختلف تعبیه می‌شوند، به کار می‌روند.

واحد یادگیری ۱: انتخاب میکروکنترلر

آیا تا به حال فکر کرده‌اید

- وقتی دکمه فراخوانی آسانسور را می‌زنید، چطور دستگاه تشخیص می‌دهد که شما در کدام طبقه هستید و کجا می‌خواهید بروید؟
 - وقتی لباسشویی شروع به کار می‌کند، چطور می‌داند که چه مقدار آب بریزد، چند بار گردش کند و چه موقع توقف کند؟
 - وقتی نزدیک درب برقی فروشگاه می‌شوید، چطور درب تشخیص می‌دهد که کسی قصد ورود دارد و خودش باز شود؟
 - وقتی دکمه ریموت تلویزیون را فشار می‌دهید، چطور تلویزیون می‌فهمد و روشن شده یا کانال عوض می‌شود؟
- پاسخ همه اینها یک چیز است: همه این دستگاه‌ها یک «مغز کوچک و هوشمند» دارند که به آن میکروکنترلر می‌گویند. میکروکنترلر مانند یک کامپیوتر کوچک است، با این تفاوت که فقط برای یک وظیفه خاص طراحی شده و پردازنده، حافظه و ورودی/خروجی‌ها را درون یک تراشه جای داده است و نیازی به سیستم عامل پیچیده ندارد. با نوشتن یک سری دستورالعمل (همان برنامه)، می‌توان به میکروکنترلر گفت که چه کاری انجام دهد. این دستورالعمل‌ها معمولاً با یک زبان برنامه‌نویسی نوشته می‌شوند که بعداً به زبان داخلی میکروکنترلر تبدیل می‌شوند. میکروکنترلر می‌تواند از ورودی‌ها (مثل صفحه کلید یا یک حسگر) اطلاعات دریافت کند و خروجی‌هایی (مثل LED، موتور یا نمایشگر) را کنترل کند.
- در مقایسه با گیت‌ها یا آی‌سی‌های دیجیتال می‌توان میکروکنترلر را یک آی‌سی هوشمند و قابل برنامه‌ریزی در نظر گرفت، یعنی قطعه‌ای که رفتار آن کاملاً به برنامه‌ای که برای آن نوشته می‌شود بستگی دارد.



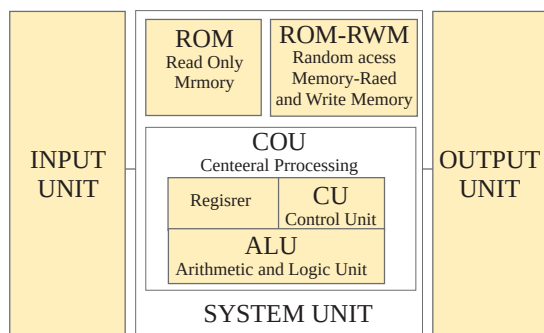
بدیهی است برای تحلیل، بررسی و اجرای عملی یک مدار کاربردی، رعایت نکات ایمنی و بهداشتی در تمامی مراحل کار ضروری است. همچنین توجه به مهارت‌های غیرفنی مانند کارگروهی، مسئولیت‌پذیری و پایبندی به نظم و ترتیب از مواردی است که اهمیت ویژه‌ای دارد و باید رعایت شود.

استاندارد عملکرد

تشریح ساختار کلی میکروکنترلر، انتخاب ESP32، یادگیری زبان برنامه‌نویسی و نوشتن برنامه ساده در محیط توسعه مرتبط

معرفی میکروکنترلرها و شرح عملکرد آنها

سیستم کامپیوتری



شکل ۱- ساختار کلی یک سامانه رایانه (کامپیوتر)

اجزای تشکیل دهنده یک سامانه کامپیوتری به صورت بلوک دیاگرام شکل ۱ نشان داده شده است. در این ساختار، CPU (واحد پردازش مرکزی) دستورات را اجرا می‌کند، ROM حافظه فقط-خواندنی برای ذخیره برنامه‌های ثابت است، RAM حافظه قابل خواندن و نوشتن برای داده‌های موقت بوده، و واحدهای ورودی/خروجی (I/O) ارتباط سیستم با محیط خارجی را برقرار می‌کنند.

واژه‌های داخل تصویر را به فارسی ترجمه کنید.

ترجمه کنید



تعریف میکروکنترلر و کاربردها

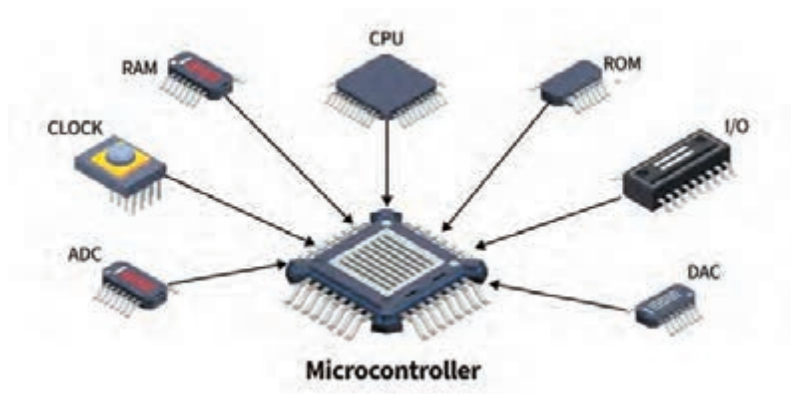
همان‌طور که می‌دانید آی‌سی‌های دیجیتال مانند HC7400 (گیت NAND) یا HC7408 (گیت AND) رفتار ثابتی دارند و تغییری در عملکرد آنها بدون تعویض فیزیکی مدار امکان‌پذیر نیست. اما اگر بخواهیم یک آی‌سی را طوری طراحی کنیم که با تغییر چند خط دستورالعمل، رفتارش عوض شود باید از میکروکنترلر استفاده کنیم.

میکروکنترلر، یک تراشه هوشمند است که درون خود پردازنده (CPU)، حافظه (معمولاً Flash برای ذخیره برنامه و RAM برای داده‌های موقت) و پورت‌های ورودی/خروجی (I/O) را در یک بسته واحد ادغام کرده است.

برخلاف آی‌سی‌های دیجیتال مانند ۷۴۰۸ که همواره به‌عنوان گیت AND عمل می‌کند، در میکروکنترلر نقش هر پایه تا حدی که سخت‌افزار داخلی اجازه دهد، قابل تغییر است، یعنی می‌توانید در یک لحظه یک پایه را به عنوان ورودی (برای خواندن وضعیت دکمه) و در لحظه‌ای دیگر همان پایه را به عنوان خروجی (برای روشن کردن LED) تعریف کنید و حتی عملکردهای پیچیده‌تری مانند ارسال داده با پروتکل سریال، خواندن سنسور آنالوگ، یا کنترل موتور را به آن تخصیص دهید.

این انعطاف‌پذیری از طریق نوشتن یک سری دستورالعمل به دست می‌آید که به آن برنامه می‌گویند. برای نوشتن برنامه از زبان‌های مختلفی مانند اسمبلی، C، پایتون و زبان‌های دیگر استفاده می‌شود. برخلاف میکروپروسسور که فقط شامل یک هسته پردازشی است و نیاز به حافظه و واحدهای I/O خارجی دارد، میکروکنترلر به‌تنهایی برای کار آماده است.

شکل ۲ ساختمان داخلی یک میکروکنترلر را نشان می‌دهد. در این شکل میکروکنترلر به عنوان یک تراشه مرکزی نشان داده شده است که با ادغام CPU، حافظه، واحدهای I/O و مدارهای تخصصی مثل ADC (تبدیل آنالوگ به دیجیتال)، DAC (تبدیل دیجیتال به آنالوگ)، و تولیدکننده پالس ساعت (Clock)، یک سیستم کامل را در یک تراشه فراهم می‌کند. این ادغام، باعث می‌شود میکروکنترلرها در کاربردهایی مانند آسانسور، لباسشویی، دستگاه‌های پزشکی و IoT که نیازمند کوچکی، مصرف انرژی پایین و قابلیت برنامه‌ریزی هستند، انتخاب اول باشند.



شکل ۲- ساختار داخلی یک میکروکنترلر

اجزای مختلف یک میکروکنترلر را با یک کامپیوتر شخصی (PC) مقایسه کنید.

پرسش



چرا در یک لباسشویی از میکروکنترلر استفاده می‌شود، نه از یک کامپیوتر معمولی؟

بارش فکری



■ اجزای داخلی میکروکنترلر

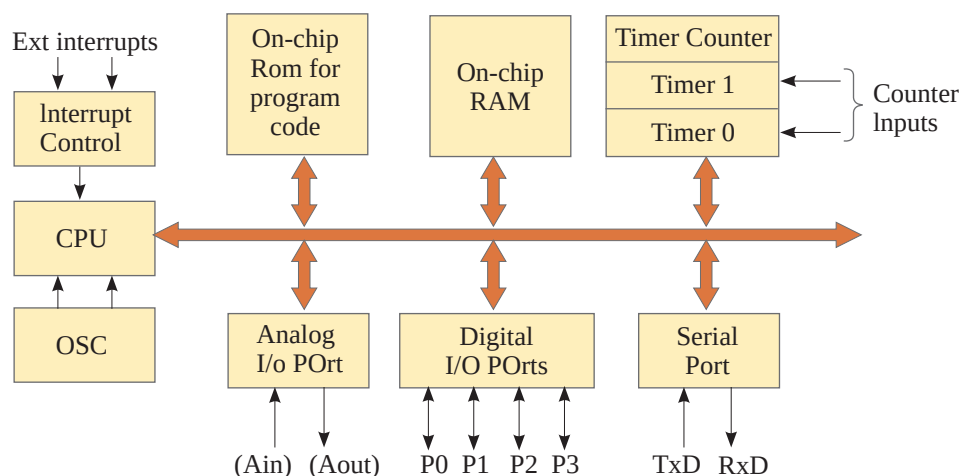
هر میکروکنترلر از چند بخش اصلی تشکیل شده است که همکاری آنها، امکان کنترل هوشمند را فراهم می‌کند.

در شکل ۳ اجزای مختلف یک میکروکنترلر به صورت بلوک دیاگرام نشان داده شده است. برخی از اجزای مهم میکروکنترلر عبارتند از:

■ **پردازنده (CPU):** مغز میکروکنترلر است که دستورالعمل‌های برنامه را یکی یکی اجرا می‌کند. عملکرد آن مانند CPU رایانه است ولی برای کارهای اختصاصی طراحی شده و به سیستم عامل نیاز نداشته و کم‌مصرف‌تر و ساده‌تر است.

■ **حافظه RAM:** فضایی موقت برای نگهداری داده‌های در حال پردازش (مثل مقدار یک سنسور یا نتیجه یک محاسبه) است. با قطع برق، محتوای حافظه RAM پاک می‌شود.

■ **حافظه Flash:** محلی است که دستورات برنامه نوشته‌شده (مثلاً به زبان C) به صورت دائمی ذخیره می‌شوند و حتی با قطع برق هم از بین نمی‌روند. این حافظه نقش حافظه برنامه را دارد. باید توجه داشت که Flash از نظر فنی با ROM متفاوت است، اما در بیشتر موارد به جای Flash از واژه ROM استفاده می‌شود.



شکل ۳- بخش‌های اصلی میکروکنترلر به صورت بلوک دیاگرام

■ **پورت‌های ورودی/خروجی (I/O):** پایه‌هایی که میکروکنترلر از طریق آنها با دنیای خارج ارتباط برقرار می‌کند. این پایه‌ها را می‌توان به صورت نرم‌افزاری به عنوان ورودی (برای خواندن کلید یا سنسور) یا خروجی (برای روشن کردن LED یا موتور) تعریف کرد.

■ **مبدل آنالوگ به دیجیتال (ADC):** بخشی است که سیگنال‌های پیوسته (مثل ولتاژ یک پتانسیومتر یا مقدار دمای نشان داده شده توسط سنسور LM35) را به اعداد دیجیتال تبدیل می‌کند.

■ **ساعت (Clock):** مانند ضربان قلب، سرعت اجرای دستورات را تعیین می‌کند، مثلاً یک میکروکنترلر ۱۶ مگاهرتز، هر ثانیه ۱۶ میلیون عمل انجام می‌دهد.

■ خودآرزیابی:

۱ فرض کنید می‌خواهیم یک دماسنج دیجیتال بسازیم که دما را روی یک نمایشگر نشان دهد. به نظر شما کدام اجزای داخلی میکروکنترلر در این پروژه استفاده می‌شود؟ دلیل آن را بنویسید.

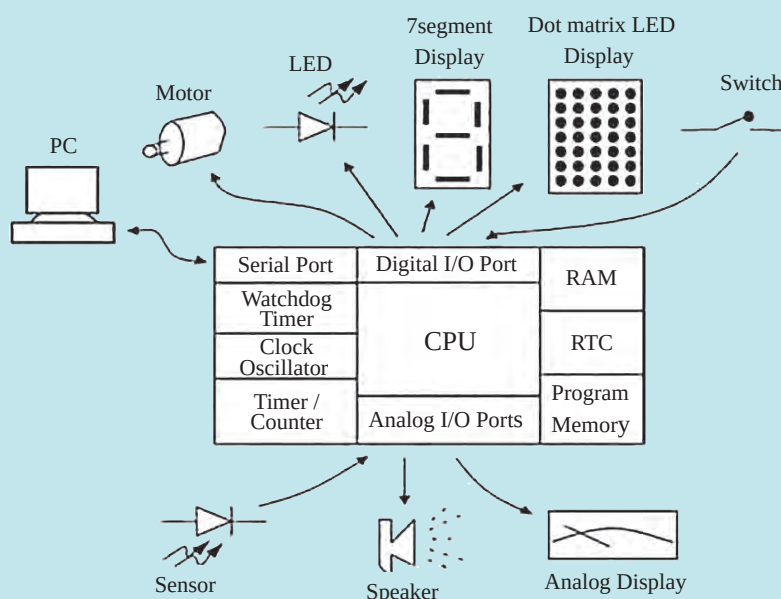
۲ حافظه Flash و RAM چه تفاوتی دارند؟ هر کدام چه زمانی استفاده می‌شود؟



ترجمه اطلاعات فنی و بخش‌های مختلف میکروکنترلر

هدف: کسب مهارت در شناخت و تشخیص اجزای مختلف یک میکروکنترلر از یکدیگر
مواد، ابزار و تجهیزات: لوازم التحریر - فرهنگ لغت انگلیسی به فارسی
مراحل اجرای کار:

۱ با بررسی بلوک دیاگرام شکل ۴، قسمت‌های اصلی و مهم آی سی میکروکنترلر را ترجمه کرده و جدول ۱ را کامل کنید.



شکل ۴- ساختار کلی یک میکروکنترلر

جدول ۱

عنوان بلوک	شماره بلوک	ترجمه
Digital I/O Port	۱	
Analog I/O Ports		
RAM (Random Access memory)		
RTC (Real Time Clock)		
Program Memory		
Serial Port		
Watchdog Timer		
Clock Oscillator		
Timer /Counter		

۲ با توجه به شکل ۴ قطعات جانبی که به درگاه‌های ورودی خروجی میکروکنترلر اتصال داده می‌شوند را تعیین کنید و در جدول ۲ بنویسید

جدول ۲

ردیف	نام دستگاه یا قطعه	ورودی برای میکروکنترلر	خروجی برای میکروکنترلر
۱	Personal Computer (PC)	به پورت سریال میکروکنترلر متصل می‌شود	
۲			
۳			
۴			
۵			
۶			
۷			
۸			
۹			

ویژگی‌های ساختاری و عملکردی میکروکنترلرها

یکی از مهم‌ترین معیارهای انتخاب میکروکنترلر مناسب برای یک پروژه، معماری و پارامترهای عملکردی اجزای داخلی میکروکنترلر (CPU، Flash، RAM، Clock و GPIO) می‌باشد. به عبارت دیگر باید بدانیم مثلاً چه میزان حافظه RAM برای یک پروژه IoT لازم است یا اینکه آیا فرکانس بالاتر همیشه بهتر است؟ در ادامه، مهم‌ترین معیارهای مقایسه را بررسی می‌کنیم.

■ **معماری بیتی** نشان‌دهنده عرض رجیسترهای عمومی پردازنده و واحد حساب و منطق (ALU) است، یعنی بیشترین تعداد بیتی که واحد پردازش مرکزی (CPU) می‌تواند در یک عملیات منطقی یا ریاضی (مانند جمع، تفریق یا مقایسه) در یک چرخه دستورالعمل یا تعداد ثابتی چرخه پردازش کند. به عنوان مثال یک میکروکنترلر ۸ بیتی (مانند ATmega32) می‌تواند در یک دستور، تا عدد $2^8 - 1 = 255$ را مستقیماً پردازش کند. اگر بخواهیم اعداد بزرگ‌تر مثل ۱۰۰۰ را پردازش کنیم، پردازنده مجبور است آن را به بخش‌های کوچک‌تر تقسیم کند و در چند مرحله محاسبه را انجام دهد، یعنی در محاسبات سنگین، سرعت میکروکنترلر کمتر شده و مصرف توان CPU بیشتر خواهد شد. ولی میکروکنترلر ۳۲ بیتی (مانند STM32F103) می‌تواند تا عدد $2^{32} - 1$ یعنی ۴'۲۹۴'۹۶۷'۲۹۵ را در یک دستور پردازش کند. در نتیجه سرعت این میکروکنترلر بسیار بالاتر بوده و برای برنامه‌های پیچیده‌تر مناسب می‌باشد.

به طور کلی میکروکنترلرهای ۸ بیتی بیشتر برای کارهای ساده و کنترلی که هزینه پایین تر و مصرف انرژی کمتری دارند مناسب هستند و میکروکنترلرهای ۳۲ بیتی برای سیستم‌های پیشرفته‌تر مانند نمایشگرهای گرافیکی، پردازش سیگنال، IoT و رباتیک که به سرعت و قدرت پردازش بالاتری نیاز دارند، مناسب هستند.

■ **فرکانس کاری (Clock Frequency)** به طور ساده تعداد دستورالعمل‌هایی است که میکروکنترلر در یک ثانیه می‌تواند اجرا کند و معمولاً با واحد Hz مانند MHz بیان می‌شود. لازم به ذکر است که فرکانس بالاتر همیشه بهتر نیست، چون توان بیشتری مصرف شده و در نتیجه گرمای بیشتری تولید می‌شود.

■ **حافظه فلش (Flash Memory)** محلی است که برنامه در آن ذخیره می‌شود و با توجه به ساده یا پیچیده بودن برنامه ممکن است از چند کیلو بایت تا چند مگا بایت مورد نیاز باشد.

■ **حافظه RAM** محل ذخیره موقت متغیر است و هر چه پروژه پیچیده‌تر باشد نیاز به حافظه RAM بیشتری است.

■ **تعداد و نوع ورودی/خروجی‌ها (GPIO)** که تعیین می‌کند چند ورودی یا خروجی مانند سنسور، رله، LED و ... را می‌توانیم به میکروکنترلر وصل کنیم.

انواع میکروکنترلرها

در جدول ۳ برخی از معروف‌ترین و پرکاربردترین میکروکنترلرها نشان داده شده است

جدول ۳- مقایسه مشخصات فنی میکروکنترلرهای پرکاربرد (از ۸۰۵۱ تا RP2040)

خانواده میکروکنترلر	نام میکروکنترلر	سال تولید	معماری بیتی	فرکانس کاری MHz	حافظه Flash	حافظه RAM	تصویر
8051	AT89S52	1995	8	33	8KB	256B	
Holtek	HT46R22	1998	8	16	4 KB	128 B	
PIC	PIC-16F877A	2000	8	20	8 KB	368 B	

	2 KB	32 KB	16	8	2002	ATmega32	AVR
	20 KB	64 KB	72	32	2007	STM-32F103C8T6	ARM
	520 KB	4 MB	240	32	2016	ESP32	ESP32
	264 KB	2 MB	133	32	2021	Raspberry Pi Pico	RP2040

در میکروکنترلرهای ESP32 و RP2040 حافظه Flash در داخل هسته پردازنده قرار ندارد و معمولاً به صورت تراشه جداگانه (خارجی) روی برد توسعه قرار می‌گیرد.

نکته مهم



بردهای توسعه میکروکنترلرها

در کنار میکروکنترلر خام، برای تسهیل فرایند توسعه و آزمایش، از بردهای توسعه (Development Boards) استفاده می‌شود. در این بردها، میکروکنترلر همراه با مدارهای جانبی ضروری از جمله تنظیم‌کننده تغذیه (رگولاتور)، مبدل USB به سریال (برای برنامه‌ریزی)، نوسان‌ساز کریستال (برای تولید سیگنال ساعت) و آرایه‌ای از پین‌های دسترسی قابل استفاده روی یک برد مدار چاپی (PCB) یکپارچه قرار می‌گیرد. این طراحی، امکان شروع سریع برنامه‌نویسی، تست و نمونه‌سازی اولیه را بدون نیاز به طراحی سخت‌افزاری پیچیده فراهم می‌کند تا بدون نیاز به ساختن مدارهای پایه، برنامه‌نویسی و آزمایش طراحی‌ها سریع‌تر انجام شود. بردهایی مانند Arduino Uno ، Blue Pill ، ESP32 DevKit و Raspberry Pi Pico نمونه‌های رایج از بردهای توسعه میکروکنترلرها هستند.

در ادامه به معرفی چند نمونه برد توسعه میکروکنترلر می‌پردازیم:



شکل ۵- برد Arduino Uno

۱ Arduino Uno

میکروکنترلر به کاررفته: ATmega328P (۸ بیتی)

ویژگی های اصلی:

- فرکانس کاری ۱۶ مگاهرتز
- حافظه Flash: ۳۲ کیلوبایت، RAM: ۲ کیلوبایت
- ۱۴ پین دیجیتال ورودی/خروجی، ۶ پین آنالوگ
- محیط برنامه نویسی ساده Arduino IDE

۲ Blue Pill

میکروکنترلر به کاررفته: STM32F103C8 (۳۲ بیتی)

ویژگی های اصلی:

- فرکانس کاری حدود ۷۲ مگاهرتز
- حافظه Flash: ۶۴ کیلوبایت، RAM: ۲۰ کیلوبایت
- ۳۷ پین دیجیتال ورودی/خروجی
- قابلیت های SPI، I2C، UART، ADC، PWM
- قیمت بسیار پایین و اندازه کوچک



شکل ۶- برد توسعه Blue pill

۳ ESP32 DevKit/Wroom

میکروکنترلر به کاررفته: ESP32-D0WD-Q6 دو هسته ای

(۳۲ بیتی)

ویژگی های اصلی:

- فرکانس کاری تا ۲۴۰ مگاهرتز
- حافظه Flash: معمولاً ۴-۸ مگابایت، RAM حدود ۵۲۰ KB
- دارای Wi-Fi و Bluetooth (Classic + BLE) داخلی
- قیمت مناسب و حجم کوچک

ESP32 گزینه ای عالی برای پروژه های پیشرفته و هوشمند است و با محیط های Arduino IDE، ESP-IDF و MicroPython قابل برنامه نویسی است.



شکل ۷- برد توسعه Esp32 Wroom

۴ Raspberry Pi Pico

میکروکنترلر به کاررفته RP2040: (۳۲ بیتی)

ویژگی های اصلی:

- فرکانس کاری ۱۳۳ مگاهرتز
- حافظه RAM داخلی: ۲۶۸ کیلوبایت
- حافظه Flash خارجی روی برد: معمولاً ۲ مگابایت
- تعداد زیادی پین دیجیتال
- قیمت بسیار پایین و اندازه کوچک



شکل ۸- برد توسعه Raspberry Pi Pico

خود ارزیابی

- ۱ میکروکنترلر چیست و چه تفاوتی با کامپیوتر دارد؟
- ۲ معماری ۸ بیتی و ۳۲ بیتی چه تفاوتی در پردازش داده‌ها ایجاد می‌کند؟
- ۳ حافظه Flash و RAM چه کاربردی در میکروکنترلر دارند؟
- ۴ چه چیزی باعث می‌شود یک میکروکنترلر مناسب پروژه IoT باشد؟
- ۵ چرا در بعضی پروژه‌ها استفاده از برد توسعه مثل Arduino یا ESP32 نسبت به استفاده مستقیم از میکروکنترلر خام آسان‌تر است؟
- ۶ میکروکنترلرها همیشه دارای سیستم‌عامل هستند.
☐ درست ☐ نادرست
- ۷ یک میکرو ۳۲ بیتی می‌تواند اعداد بزرگ‌تر را نسبت به ۸ بیتی در یک مرحله پردازش کند.
☐ درست ☐ نادرست
- ۸ بُرد Arduino Uno یک بُرد توسعه بر پایه میکروکنترلر ۳۲ بیتی است.
☐ درست ☐ نادرست
- ۹ اگر بخواهید پروژه‌ای با LED، موتور و سنسور دما بسازید، کدام میکروکنترلر ۸ بیتی یا ۳۲ بیتی مناسب‌تر است؟ چرا؟
- ۱۰ تفاوت Arduino Uno و ESP32 DevKit را با توجه به پورت‌های ارتباطی و وای‌فای توضیح دهید.
- ۱۱ چرا میکروکنترلرهای ۳۲ بیتی در پروژه‌های بلادرنگ و پردازش داده‌های سنگین مناسب‌تر هستند؟

استخراج و ترجمه اطلاعات فنی بردهای توسعه

هدف: کسب مهارت در ارتباط با استخراج اطلاعات مختلف از برگه اطلاعات بردهای توسعه متداول مواد، ابزار و تجهیزات: لوازم التحریر - فرهنگ لغات - برگه اطلاعات بردهای توسعه ESP32wroom30pin،

Blue Pill، Raspberry Pi Pico و Arduino Uno

مراحل اجرای کار:

- ۱ با مراجعه به برگه اطلاعات بردهای توسعه مشخصات کلی آنها را بررسی و جدول ۴ را کامل کنید.

کار عملی ۲



جدول ۴

Raspberry Pi Pico	Blue Pill	Arduino Uno	Esp32 Wroom 30pin	
				فرکانس کاری
				ولتاژ تغذیه
				معماری بیتی
				حافظه flash
				حافظه RAM
				تعداد کل پایه ها
				تعداد ورودی های آنالوگ
				ورودی تاچ
			دارد	Wifi داخلی
			دارد	بلوتوث داخلی

انتخاب یک میکرو کنترلر مناسب برای انجام پروژه های گوناگون

انتخاب میکروکنترلر، مانند انتخاب ابزار مناسب برای یک کار است. نمی توان برای همه پروژه ها از یک میکروکنترلر یکسان استفاده کرد. بلکه باید با توجه به نوع پروژه، حجم برنامه، تعداد ورودی و خروجی مورد نیاز، استفاده از ارتباط بی سیم، میزان مصرف انرژی، بودجه و همچنین در دسترس بودن منابع آموزشی، میکروکنترلری را انتخاب کرد که بیشترین تطابق را با اهداف پروژه داشته باشد. به طور کلی، سه شاخص مهم در انتخاب میکروکنترلر وجود دارد:

۱ تطابق فنی میکروکنترلر با نیازهای پروژه:

اولین و فنی ترین شاخص، میزان توانایی میکروکنترلر در انجام وظیفه مورد نظر با هزینه ای مناسب است. در تحلیل یک پروژه مبتنی بر میکروکنترلر، ابتدا باید مشخص شود که آیا یک میکروکنترلر ۸ بیتی یا ۳۲ بیتی می تواند نیاز پردازشی پروژه را به خوبی برآورده کند یا خیر. در این زمینه عوامل زیر اهمیت دارند:

■ **سرعت پردازنده:** حداکثر سرعت کاری میکروکنترلر چقدر است؟

■ **نوع بسته بندی:** آیا میکروکنترلر در بسته بندی هایی مانند DIP یا SMD ارائه می شود؟ این موضوع در مونتاژ، استفاده روی بردبرد و ساخت محصول نهایی اهمیت دارد.

■ **مصرف انرژی:** این عامل به ویژه در پروژه هایی که با باتری کار می کنند بسیار مهم است.

- **حافظه:** مقدار حافظه برنامه و داده باید متناسب با حجم پروژه باشد.
- **تعداد پایه‌های ورودی و خروجی و تایمرها:** این موضوع تعیین می‌کند چه تعداد حسگر، کلید یا عملکرد را می‌توان به میکروکنترلر متصل کرد.
- **امکانات جانبی:** مانند ورودی و خروجی آنالوگ، تایمرها، و پشتیبانی از پروتکل‌های ارتباطی مختلف، بلوتوث و WiFi.
- **قیمت:** قیمت هر واحد میکروکنترلر در هزینه نهایی محصول نقش مهمی دارد.

۲ سهولت توسعه و تولید محصول:

دومین شاخص مهم، میزان سهولت استفاده از میکروکنترلر در فرایند طراحی و تولید است. عواملی مانند در دسترس بودن محیط برنامه‌نویسی، وجود کامپایلر مناسب، امکان شبیه‌سازی، پشتیبانی فنی و فراوانی منابع آموزشی (کتاب، فیلم آموزشی و مثال‌های عملی) در این بخش اهمیت دارند. بسیاری از پروژه‌ها نه به دلیل ضعف فنی، بلکه به دلیل نبود منابع آموزشی مناسب با مشکل مواجه می‌شوند.

۳ **در دسترس بودن در بازار:** سومین شاخص، موجود بودن میکروکنترلر در بازار در زمان حال و آینده است. انتخاب میکروکنترلری که به‌سختی پیدا می‌شود یا احتمال توقف تولید آن وجود دارد، می‌تواند ادامه پروژه یا تولید انبوه را با مشکل روبه‌رو کند.

با در نظر گرفتن این عوامل و متناسب با نوع پروژه، می‌توان بهینه‌ترین میکروکنترلر را انتخاب کرد. برای مثال: اگر هدف ساخت یک سیستم ساده و کم‌مصرف مانند دماسنج دیجیتال یا کنترل‌کننده روشنایی باشد، میکروکنترلرهای ۸ بیتی مانند PIC یا Holtek به دلیل مصرف توان پایین و قیمت مناسب، گزینه‌های اقتصادی و کاربردی هستند.

■ اگر پروژه به پردازش سریع نیاز داشته باشد، مانند پردازش سیگنال صوتی یا کنترل ربات‌های پیشرفته، میکروکنترلرهای ۳۲ بیتی (مانند STM32، ESP32 یا RP2040 در برد Raspberry Pi Pico) به دلیل سرعت بالا و امکانات سخت‌افزاری گسترده، انتخاب مناسب‌تری هستند.

■ در پروژه‌هایی مانند رباتیک هوشمند، خانه‌های هوشمند، کشاورزی دیجیتال و به‌طور کلی سامانه‌هایی که به ارتباط بی‌سیم نیاز دارند، میکروکنترلر ESP32 به دلیل داشتن WiFi و بلوتوث داخلی، توان پردازشی مناسب، قیمت قابل قبول و پشتیبانی آموزشی گسترده، در عمل یکی از گزینه‌های بسیار مناسب به شمار می‌رود. البته این به معنی بهترین انتخاب برای همه پروژه‌ها نیست و انتخاب نهایی باید بر اساس نیاز پروژه انجام شود.

اگر یک میکروکنترلر را به‌خوبی یاد بگیرید، یادگیری سایر میکروکنترلرها بسیار ساده‌تر خواهد بود، زیرا مفاهیم اصلی مانند پردازنده، حافظه، ورودی و خروجی و اصول برنامه‌نویسی در همه آنها مشترک است. دلیل اینکه در این کتاب، میکروکنترلر ESP32 انتخاب شده است، این است که این تراشه ترکیبی از سادگی برنامه‌نویسی در محیط Arduino، توان پردازشی یک میکروکنترلر ۳۲ بیتی و امکانات نوین اینترنت اشیا مانند WiFi، بلوتوث و به‌روزرسانی نرم‌افزاری از راه دور (OTA) را در یک بستر واحد فراهم می‌کند. این ویژگی‌ها باعث می‌شوند یادگیری میکروکنترلرها برای هنرجویان ساده‌تر، کاربردی‌تر و متناسب با نیازهای امروز و آینده صنعت باشد.

نکته مهم





فردی یا گروهی

۱ فرض کنید قرار است یک سیستم آبیاری خودکار طراحی کنید که:

- رطوبت خاک را تشخیص دهد (شکل ۹-الف)
 - اگر کمتر از ۴۰٪ بود، پمپ را ۲ دقیقه روشن کند،
 - داده‌ها را هر ۱۰ دقیقه به یک سرور اینترنتی ارسال کند.
- کدام میکروکنترلر را انتخاب می‌کنید؟ چرا؟



الف



ب

شکل ۹

۲ اگر بخواهید یک ریموت کنترل ساده تلویزیون فقط شامل دکمه‌های روشن/خاموش و کنترل حجم صدا باشد، بسازید (شکل ۹-ب)، آیا نیازی به WiFi یا بلوتوث دارید؟ کدام میکروکنترلر از نظر اقتصادی بهینه‌تر است؟

برد توسعه ESP32، قابلیت‌ها و انواع آن

در سال ۲۰۱۴، ماژول ESP8266 (شکل ۱۰) توسط شرکت چینی Espressif Systems به‌عنوان یک ماژول وای‌فای کم‌هزینه معرفی شد. در ابتدا، این ماژول عمدتاً به‌عنوان یک قطعه کمکی برای افزودن قابلیت وای‌فای به میکروکنترلرهای دیگر (ATmega32) استفاده می‌شد. اما با توجه به این که ESP8266 دارای یک پردازنده ۳۲ بیتی بوده و قابلیت اجرای برنامه‌های مستقل را دارد، بنابراین به سرعت به‌عنوان یک میکروکنترلر کامل محبوبیت یافت.



شکل ۱۰- برد توسعه

ESP8266 (NodeMCU v3)

با توجه به استقبال گسترده از این ماژول، شرکت Espressif در سال ۲۰۱۶ نسخه پیشرفته‌تری را با نام ESP32 معرفی کرد. این ماژول نه تنها جایگزینی برای ESP8266 محسوب می‌شود، بلکه یک پلتفرم یکپارچه برای سیستم‌های تعبیه‌شده هوشمند است که علاوه بر وای‌فای، از بلوتوث، پردازنده دو هسته‌ای و تعداد بالایی از پایه‌های ورودی/خروجی نیز پشتیبانی می‌کند.



شکل ۱۱- ESP32-Wroom 30 pin

■ ویژگی‌های اصلی ESP32

ESP32 (شکل ۱۱) یکی از قدرتمندترین ماژول‌های میکروکنترلری در دسترس است. مهم‌ترین امکانات و ویژگی‌های این ماژول عبارت‌اند از:

■ دو هسته (Core 0 و Core 1) با امکان کنترل جداگانه هر یک از هسته‌ها

■ فرکانس پردازشی حداکثر تا ۲۴۰ مگاهرتز

■ ماژول Wi-Fi

■ دو نوع ماژول بلوتوث داخلی شامل بلوتوث کلاسیک (ClassicBluetooth) و بلوتوث BLE (Bluetooth Low Energy)

■ انواع بردهای ESP32

■ **ESP32-WROOM-32**: ماژول استاندارد با Wi-Fi و بلوتوث در شکل ۱۲ برد ESP32-WROOM را مشاهده می‌کنید.

این برد در دو مدل ۳۰ پایه و ۳۸ پایه در بازار موجود است. حرف W در نام برد، نشان‌دهنده داشتن ماژول wifi است.



شکل ۱۲- ESP32-WROOM-32

■ **ESP32-WROVER**: با PSRAM اضافه برای پردازش تصویر، صدا و پروژه‌های سنگین WROVER از ترکیب WROOM+ OVER به معنی توسعه یافته برد WROOM است

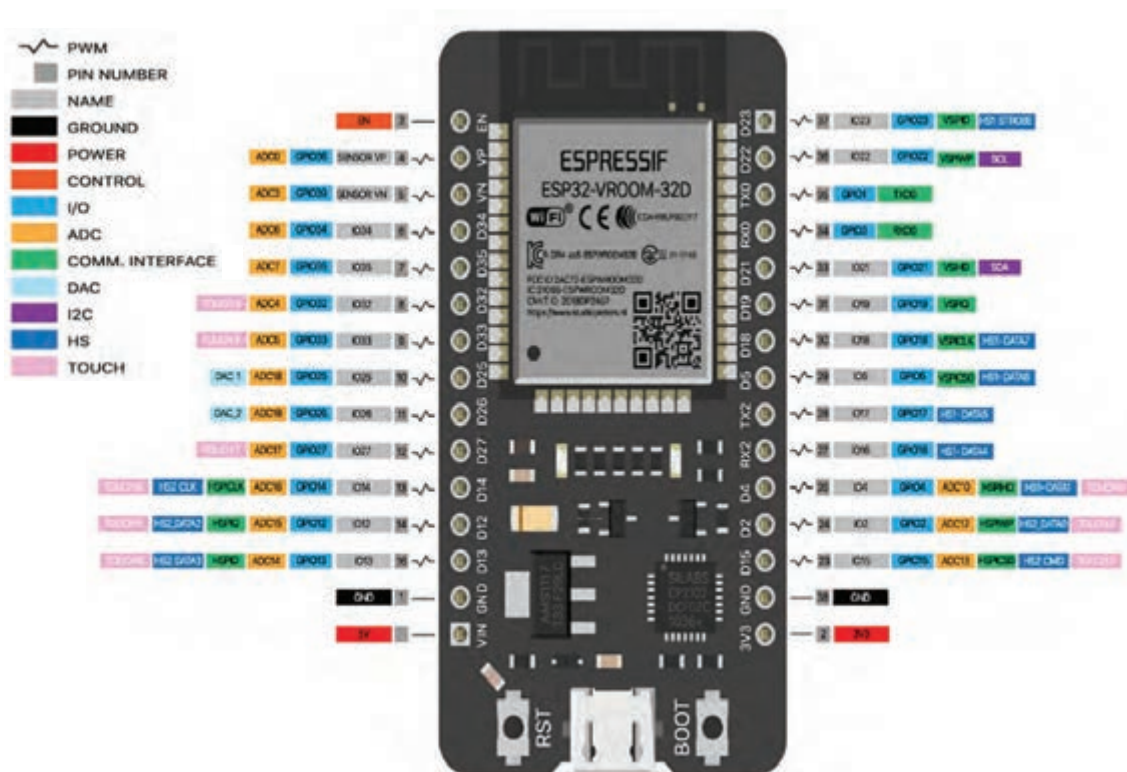
■ **ESP32-S2 و ESP32-S3**: نسخه‌های پیشرفته‌تر و امنیتی ESP32 می‌باشند که دارای قابلیت رمزنگاری و Secure Boot هستند.

در این کتاب از برد ESP32 Wroom با ۳۰ پایه استفاده می‌شود که یکی از متداول‌ترین بردهای توسعه ESP32 می‌باشد. در شکل ۱۳ پایه‌های ماژول ESP32 Wroom 30 pin به همراه نقش و عملکرد هر پایه نشان داده شده است.

در سال‌های اخیر، بردهای توسعه جدیدتری وارد بازار شده‌اند که از نظر ظاهر، پایه‌ها و عملکرد کلی مشابه بردهای ESP32-WROOM هستند، اما ممکن است نام یا شرکت مونتاژکننده متفاوتی داشته باشند. در این بردها، ماژول اصلی ESP32 همچنان همان وظیفه را انجام می‌دهد و برای اهداف آموزشی این کتاب، تفاوتی در نحوه استفاده ایجاد نمی‌شود.

نکته مهم





شکل ۱۳- پایه های برد توسعه 30 pin ESP32- wroom

هدف: آشنایی با پایه های مختلف برد ESP32 و توانایی تشخیص نوع کارکرد هر پایه
مواد، ابزار و تجهیزات: لوازم التحریر - برگه اطلاعات ESP32 Wroom
مراحل اجرای کار:

۱ با توجه به شکل شماره ۱۳ و نقشه پایه ها تعداد هر نوع پایه را مشخص کنید و جدول شماره ۵ را تکمیل کنید.

جدول ۵

شماره پایه ها	تعداد	نوع پایه
		ورودی ، خروجی دیجیتال
		PWM
		ورودی آنالوگ (ADC)
		I2c
		UART
		ورودی لمس

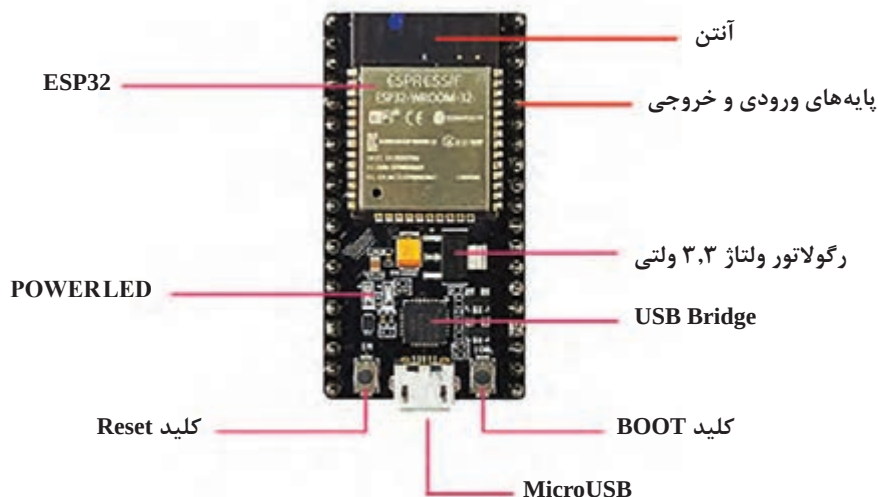
کار عملی ۳



■ اجزای برد ESP32 Wroom

در شکل ۱۴ اجزای برد توسعه ESP32 Wroom نشان داده شده است. این اجزا عبارت‌اند از:

■ **آنتن:** در ماژول‌های ESP32، آنتن مسئول ارسال و دریافت امواج رادیویی در دو پروتکل اصلی وای‌فای (2.4 GHz) و بلوتوث (BLE/Classic) است. وجود یک آنتن مناسب، شرط اولیه برقراری ارتباط بی‌سیم پایدار است.



شکل ۱۴- اجزای برد توسعه ESP32- wroom

■ **رگولاتور:** میکروکنترلر ESP32 با ولتاژ ۳٫۳ ولت کار می‌کند و تحمل ۵ ولت را ندارد. در حالی که بیشتر منابع تغذیه رایج (مثل USB یا آداپتور) ولتاژ ۵ ولت دارند، به همین دلیل روی بردهای توسعه ESP32 یک رگولاتور ولتاژ قرار داده شده است تا ولتاژ ورودی ۵ ولت را به ۳٫۳ ولت پایدار و ایمن تبدیل کند. البته توجه داشته باشید جریان خروجی این رگولاتورها بین ۱۵۰ تا ۳۰۰ میلی‌آمپر می‌باشد و برای تغذیه ماژول‌های پرمصرف نمی‌توان از آن استفاده کرد.

رگولاتور ۳ پایه دارد که پایه ورودی آن به پایه Vin برد توسعه و پایه خروجی رگولاتور به پایه ۳٫۳ V برد و پایه مشترک رگولاتور به پایه‌های GND برد توسعه متصل هستند.

■ **مبدل USB به TTL:** مبدل USB به TTL قطعه‌ای است که امکان ارتباط بین رایانه و میکروکنترلر را فراهم می‌کند. رایانه‌ها از درگاه USB استفاده می‌کنند، در حالی که میکروکنترلرها برای ارتباط سریال از سطح ولتاژ TTL بهره می‌برند. این مبدل وظیفه دارد داده‌های دیجیتال را از پروتکل USB دریافت کرده و آنها را به سیگنال‌های سریال TTL و بالعکس تبدیل کند.

در بردهای توسعه ESP32، این مبدل برای برنامه‌ریزی تراشه و همچنین ارسال و دریافت اطلاعات سریال به کار می‌رود. به کمک مبدل USB به TTL، می‌توان برنامه را از رایانه روی ESP32 بارگذاری کرد و هم‌زمان اطلاعات اجرایی برنامه را از طریق سریال مانیتور مشاهده نمود.

■ **دکمه Boot:** دکمه BOOT برای قرار دادن ESP32 در حالت دریافت برنامه استفاده می‌شود. با نگه داشتن این دکمه می‌توان برنامه را از طریق USB روی آن بارگذاری کرد. در بعضی بردهای توسعه این کار به صورت خودکار انجام می‌شود و نیازی به فشردن دکمه BOOT نیست.

■ **دکمه Reset:** دکمه ریست برای راه‌اندازی مجدد ESP32 استفاده می‌شود. با فشار دادن آن، برنامه دوباره از ابتدا اجرا می‌شود.

■ **پورت micro USB:** پورت میکرو USB محل اتصال برد ESP32 به کامپیوتر است. از این طریق، هم برنامه روی ESP32 بارگذاری می‌شود و هم برق مورد نیاز برد تأمین می‌گردد. البته در نسخه‌های جدیدتر به جای پورت micro USB از type C استفاده شده است.

■ **تغذیه و نحوه راه‌اندازی:** ولتاژ کاری ESP32 بین ۲/۲ تا ۳/۶ ولت است ولی بهترین ولتاژ برای تغذیه ۳/۳ ولت است بنابراین اگر مستقیماً از میکروکنترلر استفاده می‌شود، باید منبع تغذیه ۳/۳ ولتی فراهم شود. همانطور که در شکل ۱۳ نشان داده شده است، برد توسعه دارای دو پایه تغذیه ۳/۳ ولت و ۵ ولتی می‌باشد. بنابراین برای تغذیه این برد می‌توانیم از کابل micro USB استفاده کنیم که در این صورت می‌توان از پایه‌های ۳/۳ ولت و ۵ ولت برای تغذیه دیگر قطعات روی برد استفاده کرد و این پایه‌ها را خروجی در نظر گرفت. روش دیگر تغذیه این است که خروجی یک رگولاتور ولتاژ ۵ ولتی مثل ۷۸۰۵ را به پایه vin (5V) متصل کنیم. در این صورت پایه ۳/۳ ولت را می‌توان به عنوان خروجی گرفت و برای تغذیه قطعات جانبی مثل OLED یا RC522 که با ۳/۳ ولت کار می‌کنند استفاده کرد. در حالت سوم با یک رگولاتور ۳/۳ ولتی ولتاژ مورد نیاز ESP32 را به پایه ۳/۳ ولت متصل می‌کنیم.

■ **پین‌های نایمن در ESP32:** برخی از پین‌های ESP32 در هنگام راه‌اندازی یا بوت شدن نقش خاصی دارند و بهتر است در ابتدا از آنها استفاده نشود. در جدول صفحه بعد این پایه‌ها نشان داده شده‌اند. توجه کنید از پایه‌هایی که علامت  دارند اصلاً نباید استفاده شوند و پایه‌هایی که علامت  دارند تحت شرایط ذکر شده باید مورد استفاده قرار گیرند، مثلاً پایه ۳۴ فقط باید به صورت ورودی تعریف شود و در صورت تعریف به صورت خروجی کار نمی‌کند. جدول ۶ شماره این پایه‌ها و دلیل نایمن بودن آنها را نشان می‌دهد.

جدول ۶- پایه‌های نایمن برد توسعه Esp32

پین‌های نایمن در Esp32		
شماره پایه	دلیل	سطح نایمینی
۰	برای رفتن به حالت اجرای کد در حالت HIGH و برای برنامه‌نویسی باید LOW باشد.	
۱	برای فلش و رفع اشکال استفاده می‌شود.	
۲	باید در هنگام بوت ۰ باشد و همچنین به led آبی رنگ روی برد متصل است.	
۳	برای فلش و رفع اشکال استفاده می‌شود.	
۵	هنگام بوت شدن باید ۱ باشد.	
۶	به حافظه فلش متصل است.	
۷	به حافظه فلش متصل است	
۸	به حافظه فلش متصل است	
۹	به حافظه فلش متصل است	
۱۰	به حافظه فلش متصل است	
۱۱	به حافظه فلش متصل است	
۱۲	هنگام بوت شدن باید ۰ باشد.	
۱۵	باید در هنگام بوت ۱ باشد.	
۳۴	فقط ورودی باید تعریف شود.	
۳۵	فقط ورودی باید تعریف شود.	
۳۶	فقط ورودی باید تعریف شود.	
۳۹	فقط ورودی باید تعریف شود.	

همان‌طور که می‌دانیم عملکرد سیستم‌های دیجیتال بر پایه دو سطح منطقی صفر و یک استوار است. با ترکیب این دو حالت و با استفاده از دروازه‌های منطقی می‌توان مدارهای پیچیده‌تری مانند جمع‌کننده‌ها، مالتی پلکسر، فلیپ فلاپ، شمارنده و ... ساخت.

اما یک میکروکنترلر فقط مجموعه‌ای از گیت‌ها نیست، بلکه یک CPU کوچک درون خود دارد که می‌تواند مجموعه‌ای از دستورالعمل‌ها را اجرا کند. این دستورالعمل‌ها در نهایت به دنباله‌ای از صفر و یک (کد ماشین) تبدیل می‌شوند تا CPU بتواند آنها را پردازش کند. برای اینکه مجبور نباشیم این صفر و یک‌ها را به صورت مستقیم بنویسیم و بتوانیم دستورالعمل‌ها را ساده و قابل فهم بیان کنیم، از زبان‌های برنامه‌نویسی استفاده می‌شود. این زبان‌ها، دستورات قابل فهم برای انسان را به زبان ماشین ترجمه می‌کنند.

■ انواع زبان‌های برنامه‌نویسی با توجه به فاصله از سخت‌افزار

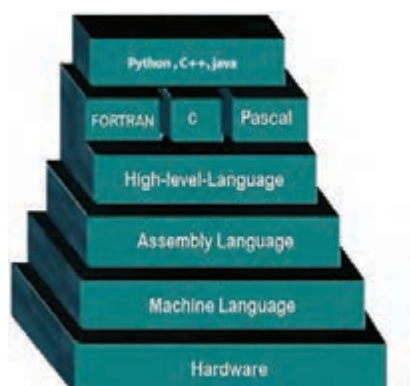
زبان‌های برنامه‌نویسی را می‌توان بر اساس نزدیکی به سخت‌افزار در سه لایه اصلی دسته‌بندی کرد: زبان ماشین، زبان اسمبلی و زبان‌های سطح بالا.

■ **زبان ماشین:** اولین و پایه‌ای‌ترین لایه، زبان ماشین است؛ این زبان تنها شکلی است که پردازنده می‌تواند مستقیماً آن را اجرا کند و به صورت دنباله‌ای از اعداد دودویی (صفر و یک) نوشته می‌شود.

■ **زبان اسمبلی:** لایه بعدی، زبان اسمبلی نام دارد و به عنوان یک نمایش خواناتر از زبان ماشین در نظر گرفته می‌شود. در این زبان، به جای توالی‌های طولانی از اعداد دودویی، از دستورات نمادینی مانند ADD, MOV استفاده می‌شود که برای انسان قابل فهم‌تر هستند.

■ **زبان سطح بالا:** لایه سوم، زبان‌های سطح بالا نام دارد. زبان‌های سطح بالا خود به دو دسته عمده تقسیم می‌شوند: زبان‌های سطح بالای کلاسیک مانند C که ساختار ساده، کارایی بالا و امکان دسترسی مستقیم به حافظه را فراهم می‌کنند و زبان‌های سطح بالای مدرن که شامل زبان‌های شیء‌گرا (مثل C++ و Java)، زبان‌های مفسری (مثل Python و JavaScript) و زبان‌های بصری (مثل Scratch) هستند و بر خوانایی، سرعت توسعه و ایمنی بیشتر تأکید دارند.

در میان تمام این زبان‌ها، زبان C جایگاه ویژه‌ای در برنامه‌نویسی سیستم‌های تعبیه‌شده دارد، چون هر چند از نظر رسمی یک زبان سطح بالا محسوب می‌شود، اما امکان دسترسی مستقیم به رجیسترهای سخت‌افزاری، عملیات بیتی و مدیریت دقیق حافظه را فراهم می‌کند. این ویژگی‌ها باعث شده است که C استاندارد طلایی برای برنامه‌نویسی میکروکنترلرها از AVR تا STM32 و ESP32 باشد. به همین دلیل، یادگیری زبان C، گامی اساسی و بسیار کارآمد برای ورود به دنیای سیستم‌های دیجیتال کنترل‌شده توسط میکروکنترلر محسوب می‌شود.



شکل ۱۵ تقسیم‌بندی زبان‌های مختلف را از نظر نزدیکی به زبان ماشین نمایش می‌دهد این نمودار به صورت ساده شده طراحی شده است و سه لایه اصلی زبان‌ها را نشان می‌دهد.

شکل ۱۵- سلسله مراتب زبان‌های برنامه‌نویسی

اولین زبان‌های برنامه‌نویسی چگونه شکل گرفتند و چگونه به زبان‌های مدرن امروز رسیده‌ایم؟
زبان‌های برنامه‌نویسی رایج دنیا کدام‌اند و هر کدام بیشتر در چه حوزه‌ای استفاده می‌شوند؟
هر زبان برنامه‌نویسی برای چه کاربردهایی مناسب‌تر است؟

تحقیق کنید



موضوع ۱: چرا زبان‌های برنامه‌نویسی سطح بالا و سطح پایین وجود دارند؟
موضوع ۲: چرا زبان‌های مختلف برنامه‌نویسی وجود دارد؟

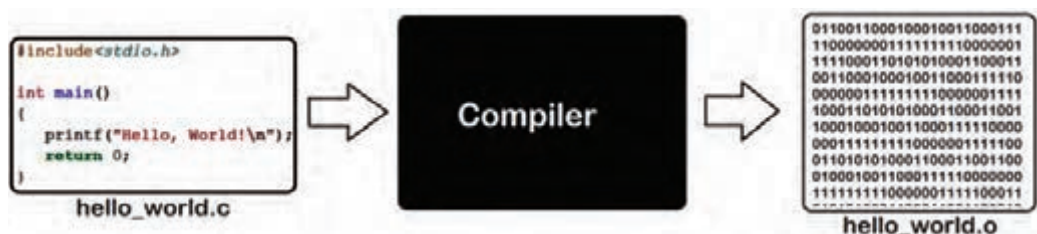
بارش فکری



■ کامپایلر، مفسر و محیط‌های توسعه (IDE)

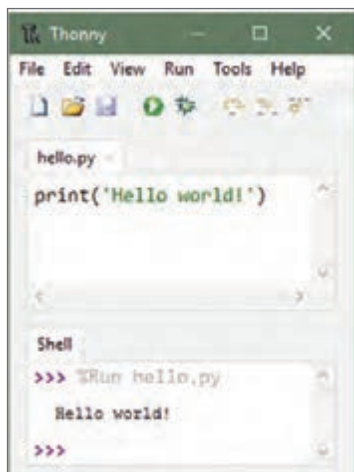
برای اینکه یک برنامه نوشته شده با زبان‌های برنامه‌نویسی بتواند روی کامپیوتر یا میکروکنترلر اجرا شود، لازم است ابتدا به زبان ماشین (صفر و یک) تبدیل شود. برای این کار بعضی از زبان‌های برنامه‌نویسی از کامپایلر و برخی دیگر از مفسر استفاده می‌کنند. در واقع کامپایلر و مفسر مترجمی هستند که کدهای برنامه را به زبان قابل فهم برای کامپیوتر (زبان ماشین) تبدیل می‌کنند.

■ **کامپایلر (Compiler):** کامپایلر یک برنامه نرم‌افزاری است که کدهای برنامه را به صورت کامل دریافت کرده و در صورت نداشتن خطا به یکباره آن را یک به کد ماشین تبدیل می‌کند. در این حالت اگر برنامه خطایی داشته باشد عمل تبدیل به کد ماشین انجام نشده و آن خطا گزارش داده می‌شود. از زبان‌های کامپایلی مهم می‌توان به زبان C، کوبول (COBOL) و C++ اشاره کرد.
در شکل ۱۶ عملکرد کامپایلر نشان داده شده است.



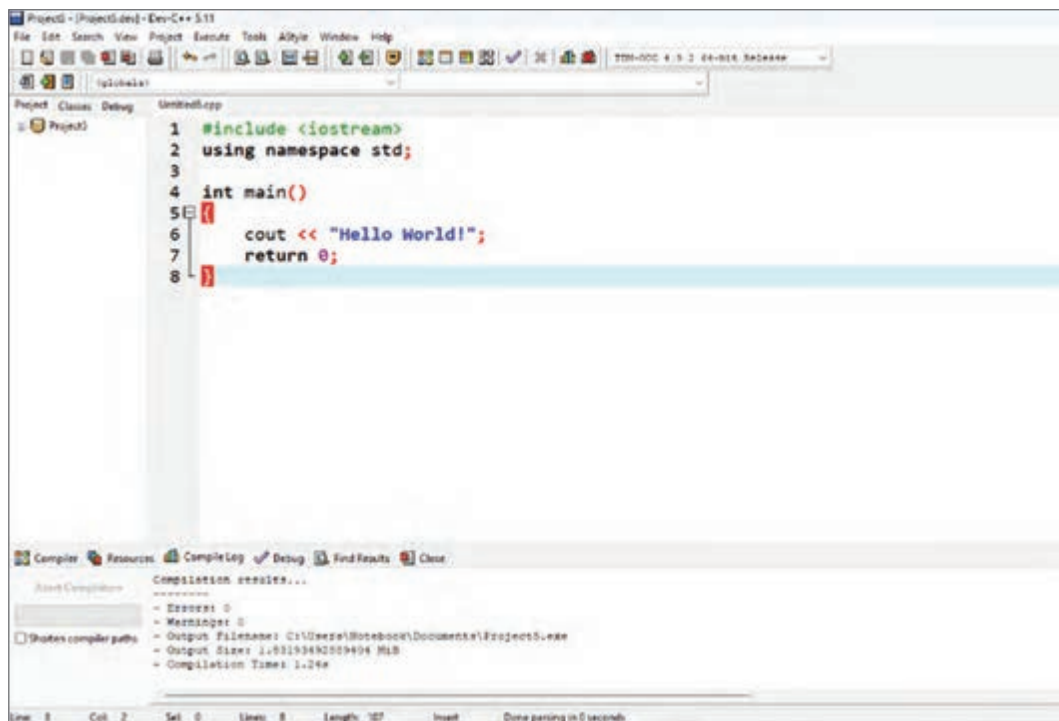
شکل ۱۶- عملکرد کامپایلر و تبدیل یک برنامه به کد ماشین

■ **مفسر (Interpreter):** مفسر یک برنامه نرم‌افزاری است که برنامه نوشته شده را خط به خط به کد ماشین تبدیل کرده و در همان لحظه اجرا می‌کند. به دلیل اینکه خطاها در حین اجرای هر خط مشخص می‌شوند، این عمل امکان رفع فوری خطاهای برنامه را امکان‌پذیر می‌سازد ولی به دلیل همزمان بودن ترجمه و اجرای برنامه، سرعت اجرای برنامه کمتر از کامپایلر است. از زبان‌های مفسری مهم می‌توان به JavaScript، Python، MATLAB اشاره کرد.



شکل ۱۷- محیط نرم‌افزار Thonny

■ **محیط توسعه یکپارچه (IDE):** IDE مخفف Integrated Development Environment بوده و یک نرم‌افزار کامل است که همه ابزارهای لازم برای برنامه‌نویسی را در یک جا قرار می‌دهد. IDE معمولاً شامل بخش‌های ویرایشگر کد (Code Editor)، کامپایلر یا مفسر، بخش نمایش خطاها، بخش اجرای برنامه، ابزارهای کمکی مثل رنگ‌بندی کد، پیشنهاد دستورات و ... می‌باشد. از IDE های معروف می‌توان به Thonny مخصوص زبان پایتون (شکل ۱۷)، Dev-C++ برای C و C++ (شکل ۱۸)، Arduino IDE برای میکروکنترلرها و Visual Studio Code اشاره کرد.



شکل ۱۸- محیط نرم‌افزار Dev-C++

خود ارزیابی

- ۱ کدام یک از گزینه‌های زیر، دسته‌بندی صحیح زبان‌های برنامه‌نویسی مدرن را نشان می‌دهد؟
 الف) زبان ماشین، زبان اسمبلی و زبان‌های سطح بالا
 ب) زبان‌های کلاسیک، زبان‌های بصری و زبان‌های ماشین
 پ) زبان‌های شیء‌گرا، زبان‌های مفسری و زبان‌های بصری
 ت) زبان‌های نمادین، زبان‌های دودویی و زبان‌های سطح بالا
- ۲ زبان ماشین تنها زبانی است که پردازنده می‌تواند مستقیماً آن را اجرا کند و به‌صورت دنباله‌ای از صفر و یک نوشته می‌شود.
☐ درست
☐ نادرست
- ۳ زبان اسمبلی به دلیل استفاده از دستورات نمادینی مانند MOV و ADD، برای انسان قابل فهم‌تر از زبان ماشین است.
☐ درست
☐ نادرست
- ۴ دو ویژگی اصلی زبان‌های سطح بالای کلاسیک مانند (C) که آنها را از زبان‌های سطح بالای مدرن متمایز می‌کند، چیست؟
- ۵ کدام یک از موارد زیر، ویژگی اصلی عملکرد «کمپایلر» را به درستی توصیف می‌کند؟
 الف) کدها را خط‌به‌خط و در حین اجرا به زبان ماشین تبدیل می‌کند.
 ب) کدها را به‌صورت کامل دریافت کرده و در صورت عدم وجود خطا، یک‌باره به کد ماشین تبدیل می‌کند.
 ج) تنها زبان‌های سطح بالا را به زبان اسمبلی ترجمه می‌کند.
 د) اگر برنامه خطا داشته باشد، آن را اصلاح کرده و سپس اجرا می‌کند.
- ۶ دو مورد از زبان‌های برنامه‌نویسی که در متن به‌عنوان مثال‌های زبان‌های کامپایلری ذکر شده‌اند، کدام‌اند؟
- ۷ کدام یک از موارد زیر کامپایلر است و کدام مفسر؟

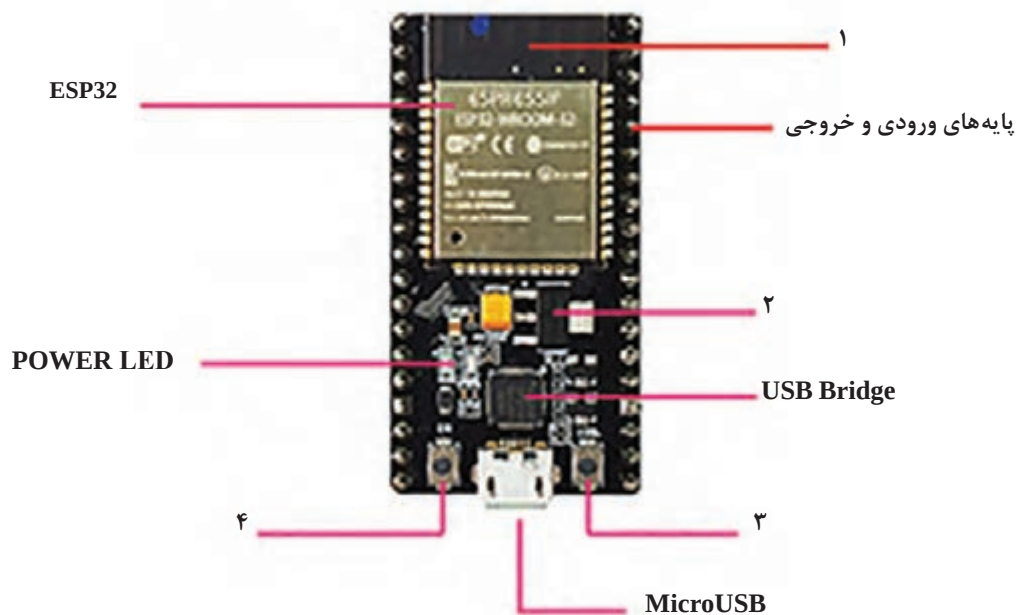
جدول ۷

عنوان	کمپایلر	مفسر
Python		
gcc		
JavaScript		
Dev-C++		

- ۸ کدام یک از موارد زیر، جزو ابزارهای کمکی و ویژگی‌های معمول یک محیط توسعه یکپارچه (IDE) نیست؟
 الف) ویرایشگر کد با قابلیت رنگ‌بندی (Syntax Highlighting)
 ب) کامپایلر یا مفسر داخلی
 پ) سخت‌افزار فیزیکی میکروکنترلر
 ت) بخش نمایش خطاها و پیشنهاد دستورات

۹ نام سه مورد از IDEهای معروفی که در متن برای زبان‌های پایتون، C/C++ و میکروکنترلرها ذکر شده‌اند را بنویسید.

۱۰ با توجه به شکل ۱۹ میکروکنترلر ESP32 Wroom، شرح دهید که هر یک از اجزای شماره‌گذاری شده (از ۱ تا ۴) چه عملکردی دارند و چه کاربردی در مدارهای پروژه دارند؟



شکل ۱۹

ارزشیابی شایستگی انتخاب میکروکنترلر

نام و نام خانوادگی		شماره ملی	تاریخ ارزشیابی:	نوبت:
کد حرفه	۸۲۱۲۱۲	حرفه:	مونتازکاران تجهیزات و وسایل برقی و الکترونیکی	کد پیمانه/پودمان
کد وظیفه شغل	۸۲۱۲۱۲۰۲	وظیفه شغل:	ساخت مدارهای دیجیتال و میکروکنترلر	سطح صلاحیت
کد کار	۸۲۱۲۱۲۰۳۴۱	کار	انتخاب میکروکنترلر	سطح شایستگی
			استاندارد عملکرد: تشریح ساختار کلی میکروکنترلر، انتخاب ESP32، یادگیری زبان برنامه‌نویسی و نوشتن برنامه ساده در محیط توسعه مرتبط	مهارت

ردیف	مرا حل کار	شرایط انجام کار (ابزار، مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره دهی)	نتایج ممکن	نمره کسب شده
۱	مبانی سخت‌افزار و انتخاب: تسلط بر بلوک دی‌گرام و توانایی انتخاب فنی میکروکنترلر (مانند ESP32) متناسب با کاربرد.	کلاس درس یا کارگاه الکترونیک مجهز به رایانه، برگه اطلاعاتی میکروکنترلرها، کاتالوگ قطعات، در زمان تعیین شده	آزمون کتبی تشریحی، پرسش و پاسخ شفاهی فنی	تشخیص دقیق بلوک‌ها و انتخاب منطقی ESP32 با توجیه فنی کامل	۳	
					۲	
					۱	
۲	راه‌اندازی محیط توسعه: نصب صحیح IDE و کامپایلر و پیکربندی اولیه بستر برنامه‌نویسی	رایانه با دسترسی اینترنت، فایل نصب IDE، درایورها زمان: ۲ ساعت مکان: آزمایشگاه کامپیوتر مواد: نرم‌افزار Arduino IDE	چک‌لیست عملی، مشاهده مستقیم فرایند نصب	نصب موفق، شناسایی پورت COM، نصب کتابخانه‌های ضروری بدون خطا نصب موفق اما نیاز به راهنمایی برای تنظیمات اولیه یا درایورها عدم توانایی در نصب صحیح یا ناتوانی در اتصال IDE به سیستم	۳	
					۲	
					۱	
۳	تسلط بر محیط توسعه و زبان برنامه‌نویسی توانایی نصب و پیکربندی محیط توسعه (IDE) و آشنایی با ساختار، سینتکس و توابع اصلی زبان برنامه‌نویسی مورد استفاده (مانند C++ برای ESP32 برنامه‌نویسی)	ابزار/ تجهیزات: برد ESP32، کابل USB، LED، مقاومت. زمان: ۳ ساعت، مکان: کارگاه الکترونیک. مواد: سورس کد پایه (Blind code) یا از روی حافظه.	بررسی کد منبع (Code Review)، تست عملکرد سخت‌افزاری، دیباگ زنده	کد تمیز، کامپایل بدون خطا، اجرای صحیح روی برد بدون کمک کد دارای ایرادات جزئی سینتکسی که خود اصلاح می‌کند یا اجرا با تأخیر خطاهای متعدد کامپایلر، عدم بارگذاری کد یا عملکرد نادرست مدار	۳	
					۲	
					۱	
۴	پیاده‌سازی، اجرا و عیب‌یابی برنامه توانایی نوشتن، کامپایل و بارگذاری برنامه‌های ساده (مانند چشمک‌زن LED یا خواندن داده سنسور) روی ESP32 و بررسی خروجی برنامه برای تشخیص و رفع خطاهای رایج نرم‌افزاری و ارتباط با سخت‌افزار.	ابزار/تجهیزات: مولتی‌متر، پروب تست، کنسول سریال IDE. زمان: ۲ ساعت، مکان: کارگاه الکترونیک. مواد: سناریوی خطای از پیش تعریف شده توسط هنرآموز	مشاهده فرایند حل مسئله، پرسش و پاسخ حین عیب‌یابی، گزارش نهایی رفع اشکال	تشخیص سریع ریشه مشکل، استفاده هوشمندانه از ابزار دیباگ و رفع کامل تشخیص مشکل اما کند بودن در فرایند عیب‌یابی یا نیاز به کمک ناتوانی در شناسایی خطا یا تکرار همان خطا پس از تلاش ناموفق	۳	
					۲	
					۱	
شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش:						
ارزشیابی کار (شایستگی انجام کار)		رعایت نسبی اصول ایمنی، بهداشت و ملاحظات زیست‌محیطی، با خطاهای جزئی در حین انجام کار.				
		عدم رعایت اصول ایمنی، بهداشت و ملاحظات زیست‌محیطی، یا بروز خطاهای جدی در نگرش و رفتار کاری.				
		☐ بلی				
		☐ خیر				

واحد یادگیری ۲: برنامه نویسی میکروکنترلر

آیا تا به حال فکر کرده‌اید

- چگونه یک میکروکنترلر می‌تواند به‌طور دقیق رفتارهای مختلف را کنترل کند؟
- چگونه یک برنامه نوشته شده روی میکروکنترلر می‌تواند باعث روشن شدن یا خاموش شدن یک LED شود؟
- چگونه تغییر زمان رنگ چراغ راهنمایی یا روشن شدن یک دستگاه می‌تواند نتیجه اجرای یک دستور ساده در یک میکروکنترلر باشد؟
- چرا میکروکنترلرها برای درک دستورات ما، به ابزارهایی نظیر بُردهای توسعه و کامپایلرها نیاز دارند؟
- چگونه می‌توان با نوشتن یک برنامه ساده، دیالوگی دقیق و منطقی میان دنیای فیزیکی و دنیای دیجیتال برقرار نمود؟

در واحد یادگیری پیشین، با مغزهای هوشمند دنیای دیجیتال، یعنی میکروکنترلرها و نقش آنها در دستگاه الکترونیکی پیرامونی خودمان آشنا شدیم. همچنین مشاهده شد که برای بهره‌گیری از این مغزها (میکروکنترلرها)، به ابزارهایی نظیر بُردهای توسعه (مانند ESP32) و کامپایلرها نیاز است تا دستورات را به زبانی تبدیل کنند که میکروکنترلر قادر به درک آن باشد. در این واحد یادگیری، هدف، نوشتن برنامه‌های ساده برای میکروکنترلر با بهره‌گیری از یک زبان برنامه‌نویسی ساختاریافته است. این کار با نوشتن یکی از ساده‌ترین برنامه‌های ممکن، یعنی برنامه‌ای برای چشمک‌زدن یک LED، آغاز می‌شود. این اقدام صرفاً یک تمرین نیست؛ بلکه دریچه‌ای به سوی جهانی است که در آن انسان و ماشین با یکدیگر دیالوگی دقیق و منطقی برقرار می‌کنند.

استاندارد عملکرد

برای برنامه‌نویسی بُرد ESP32 محیط Arduino IDE را راه‌اندازی کند و با اجرای برنامه‌های ساده ورودی‌ها و خروجی‌های بُرد را در قالب پروژه‌های عملی بتواند کنترل کند.

آشنایی با محیط برنامه‌نویسی Arduino IDE

برای برنامه‌نویسی میکروکنترلرها، از محیط‌های توسعه نرم‌افزاری (IDE) استفاده می‌شود. IDE مجموعه‌ای از ابزارهای یکپارچه است که امکان نوشتن، بررسی، کامپایل و آپلود برنامه‌ها را برای میکروکنترلر فراهم می‌کند.

یکی از محیط‌های توسعه در آموزش میکروکنترلرها، Arduino IDE است. این نرم‌افزار علیرغم ظاهری ساده، قدرت بسیار زیادی دارد. از جمله اینکه برای مبتدی‌ها ساده و روان است، از صدها بُرد مختلف (از جمله ESP32) پشتیبانی می‌کند و منابع آموزشی و نمونه کدهای فراوانی برای آن در دسترس است. در این کتاب، از Arduino IDE (نسخه ۲.۰.x) برای برنامه‌نویسی میکروکنترلر ESP32-devkit (ESP32-Wroom) استفاده می‌شود، به دلیل اینکه بهترین ترکیب از سادگی، قدرت و دسترسی را برای آموزش حرفه‌ای برای هنرجویان فراهم می‌کند.

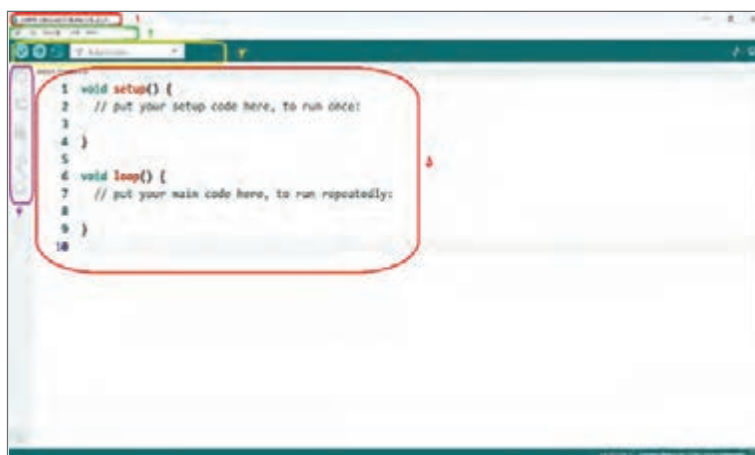
کاربرد Arduino IDE در برنامه‌نویسی میکروکنترلرها

محیط Arduino IDE امکان انجام مراحل زیر را فراهم می‌کند:

- نوشتن برنامه (کد) برای میکروکنترلر
 - بررسی و ترجمه برنامه به زبان قابل فهم برای میکروکنترلر
 - ارسال برنامه به بُرد میکروکنترلری از طریق درگاه USB بدون نیاز به پروگرامر خاص
 - مشاهده پیام‌ها و خطاهای مربوط به برنامه
- به برنامه‌هایی که در محیط Arduino IDE نوشته می‌شوند، معمولاً اسکچ (Sketch) گفته می‌شود.

معرفی محیط Arduino IDE

بعد از دانلود نرم‌افزار Arduino IDE از سایت رسمی آردوینو یا دیگر سایت‌های معتبر و اجرای آن صفحه‌ای مانند شکل ۲۰ باز می‌شود



شکل ۲۰- محیط نرم افزار Arduino IDE

بخش‌های اصلی نرم‌افزار

۱ **عنوان پنجره (Title Bar):** در بالای صفحه، نام فایل اسکچ (sketch_dec26a) و نسخه نرم‌افزار (۲.۳.۶) (Arduino IDE) نمایش داده می‌شود و نشان می‌دهد که کدام اسکچ باز است و از چه نسخه‌ای از IDE استفاده می‌کنید. (شکل ۲۱)



شکل ۲۱- نوار فهرست‌ها

۲ **نوار فهرست‌ها (Menu Bar):** که شامل منوهای اصلی زیر است (شکل ۲۲):

- **File:** برای ایجاد، ذخیره یا باز کردن اسکچ
- **Edit:** ویرایش کد (کپی، جست‌وجو، جایگزینی و غیره)
- **Sketch:** تنظیمات مربوط به کد (کامپایل، اضافه کردن کتابخانه و غیره)
- **TOOLS:** تنظیمات بُرد، پورت، سرعت ارسال و غیره که قبل از بارگذاری برنامه باید بررسی شود.
- **Help:** راهنمای نرم‌افزار و لینک به منابع آموزشی با هر یک از منوهای فوق به مرور آشنا می‌شویم.



شکل ۲۲- نوار ابزار

۳ نوار ابزار (Toolbar): دکمه‌های پرکاربرد نوار ابزار (شکل ۲۳) عبارت‌اند از:

- **Verify (✓):** بررسی صحت کد بدون بارگذاری (آپلود). اگر خطایی وجود داشته باشد، در پنجره پیام‌ها نمایش داده می‌شود.
- **Upload (→):** آپلود کد به بُرد. (هم صحت کد بررسی می‌شود و هم ترجمه می‌شود و در نهایت روی میکرو نوشته می‌شود).
- **انتخاب بُرد (Board Selector):** در این قسمت، نوع بُرد میکروکنترلری که می‌خواهید کد روی آن اجرا شود، انتخاب می‌شود. در این تصویر، Arduino Nano انتخاب شده است، اما ما در این کتاب از ESP32 Dev Module استفاده خواهیم کرد.

قبل از آپلود، حتماً این گزینه را با بُرد واقعی خود تطبیق دهید!

نکته مهم



- ### ۴ نوار ابزار کناری (Sidebar یا Tool Panel):
- این نوار شامل ابزارهای کمکی است که در مدیریت پروژه و کد، از جمله مدیریت فایل‌ها، مدیریت بُرد (نصب بُردهایی مانند ESP32)، نصب کتابخانه‌ها، دیباگ (ابزار بررسی رفتار کد در حین اجرا) و جستجو در کد کمک می‌کند.



شکل ۲۳- نوار ابزار کناری

- ### ۵ پنجره ویرایش کد (Code Editor):
- شکل ۲۴ محل نوشتن کد است که شامل دو بخش اصلی زیر می‌باشد:

- **تابع setup():** کدی که فقط یک بار در ابتدا اجرا می‌شود (مثل تنظیم پین‌ها).
- **تابع loop():** کدی که به‌طور مداوم و تکراری اجرا می‌شود و هسته عملکرد میکروکنترلر است.



شکل ۲۴- پنجره ویرایش کد

هر اسکچ حداقل باید این دو تابع را داشته باشد، حتی اگر داخلشان کدی نباشد.

نکته مهم



مفهوم دیباگ کردن چیست و چرا در برنامه‌نویسی اهمیت دارد. تحقیق کرده و نتیجه را در کلاس ارائه دهید.

تحقیق کنید



■ نصب و راه‌اندازی هسته ESP32 در Arduino IDE

برای آنکه بتوان یک میکروکنترلر را در محیط Arduino IDE برنامه‌نویسی کرد، لازم است هسته نرم‌افزاری (Core) مربوط به آن بُرد در نرم‌افزار نصب شود. هسته نرم‌افزاری مجموعه‌ای از فایل‌ها و تنظیمات است که امکان شناسایی بُرد و ارتباط با آن را فراهم می‌کند.

برای انجام عملیات نصب باید رایانه به اینترنت متصل باشد.

نکته



■ مفهوم بُرد و هسته نرم‌افزاری

در Arduino IDE، هر میکروکنترلر یا بُرد توسعه با عنوان Board شناخته می‌شود. برای هر Board، یک هسته نرم‌افزاری مخصوص وجود دارد که شامل تعاریف پایه‌های ورودی و خروجی، تنظیمات پردازنده و کتابخانه‌های مورد نیاز می‌باشد. برای استفاده از بُرد ESP32 در Arduino IDE، باید هسته مربوط به آن نصب شود.

پودمان چهارم: انتخاب میکروکنترلر (Microcontroller) و برنامه‌ریزی آن

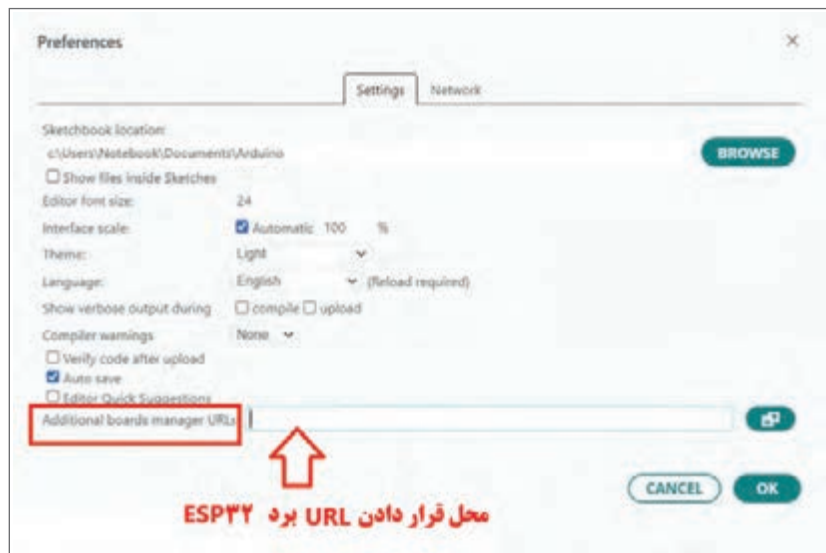
■ افزودن هسته ESP32 به Arduino IDE:

برای این کار مطابق شکل ۲۵ ابتدا از منوی file گزینه Preferences را انتخاب می‌کنیم.



شکل ۲۵- مراحل نصب برد ESP32 در آردوینو

سپس در پنجره باز شده (شکل ۲۶)، در بخش Additional Board Manager URLs لینک نصب بُرد ESP32 در آردوینو را وارد می‌کنیم. این لینک در شکل ۲۷ موجود است. همچنین می‌توان با جست‌وجوی عبارت «لینک نصب بُرد ESP32 در آردوینو» در موتورهای جستجوی اینترنتی این لینک را به‌دست آورد.



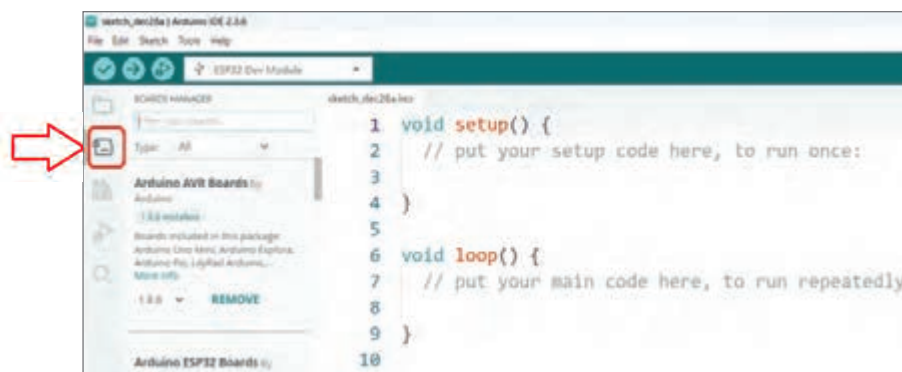
شکل ۲۷- لینک نصب برد
ESP32 در Arduino IDE

شکل ۲۶- محل قرار دادن لینک نصب برد ESP32

این لینک در آدرسی مانند زیر است:

https://ESPRESSIF.github.io/arduino-ESP32/package_ESP32_index.json

بعد از قرار دادن لینک و انتخاب گزینه OK نرم افزار شروع به دانلود موارد مورد نیاز برای نصب بُرد می کند. بعد از پایان دانلود از نوار ابزار Tool Panel (در سمت چپ نرم افزار)، گزینه Board manager را انتخاب کرده و در پنجره مربوطه بُرد ESP32 را انتخاب می کنیم. (شکل ۲۸)



شکل ۲۸- منوی Board manager در Arduino IDE

در مرحله بعد مانند شکل ۲۹ گزینه install را انتخاب کرده تا بُرد ESP32 در Arduino IDE نصب شود. پس از پایان نصب، بُردهای مختلف ESP32 به فهرست بُردهای قابل انتخاب در نرم افزار اضافه می شوند.



شکل ۲۹- نصب ESP32 در Arduino IDE

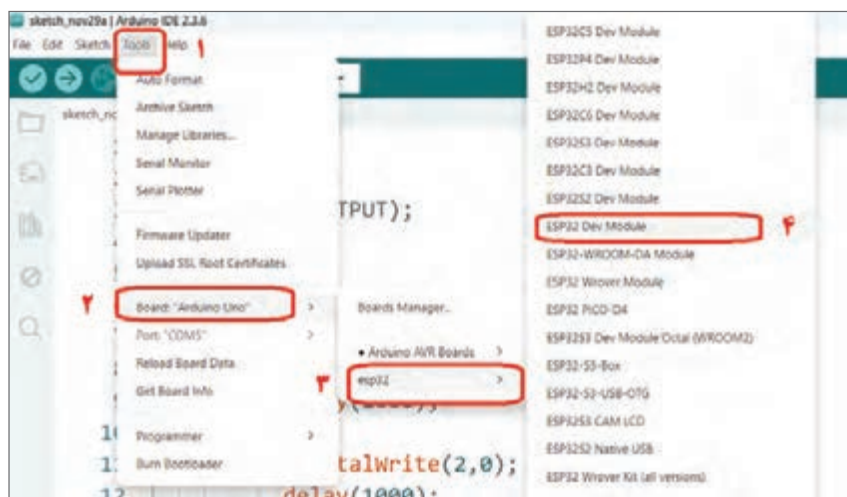
این عملیات فقط یک بار انجام می شود و پس از آن نیازی به تکرار ندارد.

نکته



انتخاب بُرد و درگاه ارتباطی

پس از نصب هسته ESP32 نوبت به انتخاب بُرد مورد نظر می‌رسد برای این کار مانند شکل ۳۰ ابتدا از منوی TOOLS گزینه Board را انتخاب می‌کنیم سپس در کشوی باز شده ESP32 را انتخاب می‌کنیم. در پنجره باز شده نام تمام بُردهای ESP32 که در مرحله قبلی در نرم‌افزار نصب شده نمایش داده می‌شود و با توجه به بُرد توسعه‌ای که تهیه کرده‌ایم یکی از گزینه‌ها را انتخاب می‌کنیم.



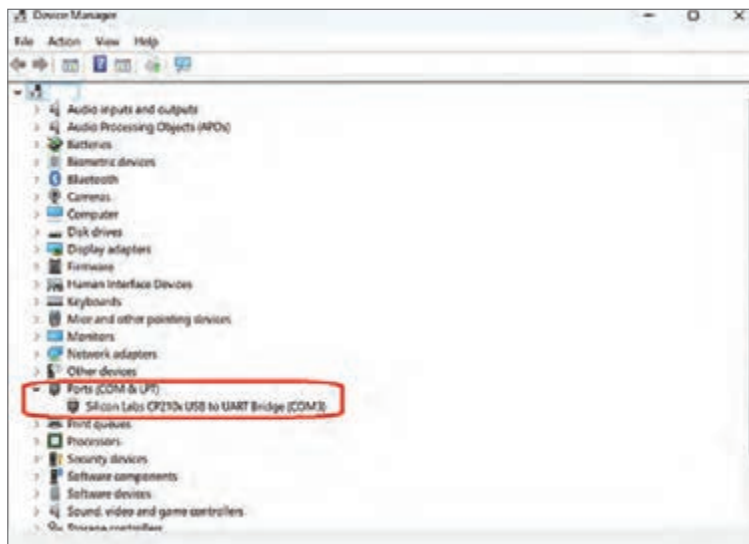
شکل ۳۰- مراحل انتخاب برد

با توجه به اینکه در این کتاب از بُرد توسعه ESP32 Wroom 30 pin و یا بُردهای جدیدتر که معادل آن هستند استفاده می‌شود، گزینه ESP32 Dev Module را انتخاب می‌کنیم. گام بعدی انتخاب درگاه ارتباطی (Port) متناسب با بُرد است. برای این کار از منوی TOOLS گزینه Port را انتخاب و پورت اختصاص داده شده به بُرد را مشخص می‌کنیم. انتخاب صحیح بُرد و پورت، نقش مهمی در موفقیت‌آمیز بودن بارگذاری (آپلود) برنامه دارد. در بخش بعدی نحوه تشخیص پورت مناسب مربوط به ESP32 گفته خواهد شد.

نصب درایور USB به Serial در ویندوز

آخرین گام برای نصب بُرد ESP32 نصب درایور بُرد در ویندوز است. در بسیاری از بُردهای ESP32، برای برقراری ارتباط بین بُرد و رایانه از تراشه‌های USB به Serial استفاده می‌شود. دو تراشه‌ی رایج در این بُردها عبارت‌اند از CP210x و CH340.

در سیستم‌عامل ویندوز، برای شناسایی صحیح این تراشه‌ها، نصب درایور مربوطه ضروری است. برای تشخیص نوع درایور و همچنین شماره پورتی که به بُرد ESP32 اختصاص داده شده، از بخش Control Panel ویندوز یا با فشردن کلیدهای Win+X، گزینه Device Manager را انتخاب کنید تا پنجره‌ای مانند شکل ۳۱ نمایش داده شود. در بخش Ports نام درایور و شماره پورت مربوطه مشخص شده است. براساس نامی که در اینجا آمده درایور مورد نظر را از اینترنت دانلود کرده و آن را در سیستم خود نصب کنید. بعد از این مرحله نصب بُرد شما تکمیل شده و آماده نوشتن و بارگذاری برنامه‌ها می‌باشد.



شکل ۳۱- شماره پورت و نام درایور برد ESP32 در Device Manager

کار عملی ۱



نصب نرم افزار Arduino IDE و بُرد توسعه ESP32

هدف: کسب مهارت در نصب نرم افزار آردوینو و همچنین نصب هسته بُرد توسعه ESP32 در نرم افزار آردوینو و در ویندوز

مواد، ابزار و تجهیزات: رایانه متصل به اینترنت، نرم افزار Arduino IDE، بُرد توسعه ESP32.
مراحل اجرای کار:

پس از مشاهده فیلم روش نصب نرم افزار آردوینو و نصب بُرد توسعه ESP32 در آردوینو و در ویندوز، این نرم افزار را نصب کنید و بُرد توسعه ESP32 را در ویندوز و نرم افزار آردوینو نصب کنید. شماره پورت اختصاص داده شده به بُرد را مشخص کنید.

نکته مهم



یکی از مشکلات متداول در هنگام ESP32 به کامپیوتر توسط کابل USB (در نسخه های قدیمی تر میکرو USB و در نسخه های جدیدتر Type C) جدا شدن پورت USB روی برد ESP32 است، در هنگام وصل و جدا کردن کابل USB حداکثر دقت را مبذول فرمایید.



شکل ۳۲- ساختار برنامه آردوینو

■ ساختار برنامه در آردوینو

برنامه هایی که در محیط Arduino IDE برای میکروکنترلر نوشته می شوند، دارای ساختاری مشخص و از پیش تعریف شده هستند (شکل ۳۲). این ساختار باعث می شود میکروکنترلر بداند کدام دستورات در ابتدای کار و کدام دستورات به صورت مداوم اجرا شوند.

هر برنامه آردوینو (اسکچ) به‌طور معمول شامل دو تابع اصلی `setup()` و `loop()` است:

■ **تابع `setup()`:** این تابع بخشی از برنامه است که فقط یک بار و در ابتدای روشن شدن یا راه‌اندازی مجدد میکروکنترلر اجرا می‌شود. از این تابع معمولاً برای کارهای اولیه مانند موارد زیر استفاده می‌شود:

□ تعیین نوع پایه‌ها (ورودی یا خروجی)

□ راه‌اندازی ماژول‌ها و تجهیزات جانبی

□ تنظیم مقادیر اولیه متغیرها

پس از اجرای کامل تابع `setup()`، میکروکنترلر وارد مرحله اجرای تابع `loop()` می‌شود.

■ **تابع `loop()`:** بخشی از برنامه است که به‌صورت پیوسته و تکراری اجرا می‌شود. تمام رفتار اصلی میکروکنترلر مانند بررسی ورودی‌ها و کنترل خروجی‌ها معمولاً در این تابع نوشته می‌شود.

به بیان ساده، می‌توان گفت:

□ دستوراتی که باید همیشه تکرار شوند، در تابع `loop()` قرار می‌گیرند.

□ پس از رسیدن به انتهای این تابع، اجرای آن دوباره از ابتدا آغاز می‌شود.

■ آشنایی با چند قانون پایه در نوشتن برنامه‌های آردوینو

زبان برنامه‌نویسی مورد استفاده در محیط Arduino IDE مبتنی بر زبان C است که به‌صورت ساده شده و مناسب برای آموزش در آمده است. برای نوشتن برنامه‌های آردوینو، قوانین و قواعد ذیل را باید رعایت کرد:

۱ در برنامه‌های آردوینو (زبان C) هر دستور باید با علامت `;` (semicolon) پایان یابد. در صورت فراموشی این علامت، برنامه با خطا مواجه می‌شود.

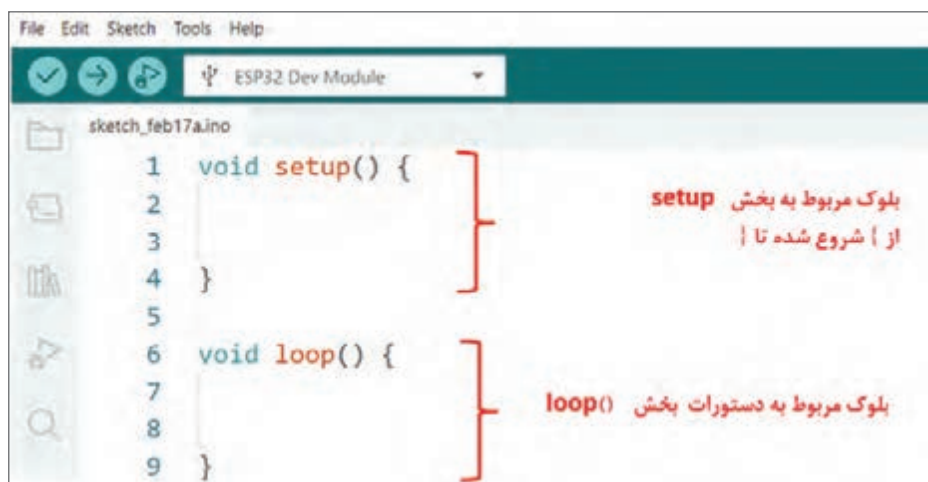
۲ زبان آردوینو به حروف بزرگ و کوچک حساس است. برای مثال، نوشتن `Loop` به‌جای `loop` یا `Setup` به‌جای `setup` باعث بروز خطا در برنامه می‌شود (شکل ۳۳).



شکل ۳۳- خطا در نوشتن حروف بزرگ

۳ برای مشخص کردن بخش‌های مختلف برنامه شکل ۳۴ مانند توابع `loop()` و `setup()` از آکولاد باز «}» و آکولاد بسته «{» استفاده می‌شود. هر آکولاد باز باید یک آکولاد بسته متناظر داشته باشد.

۴ توضیح‌نویسی در برنامه (Comment): برای توضیح بخش‌های مختلف برنامه، می‌توان از توضیح‌نویسی استفاده کرد. این توضیحات توسط میکروکنترلر اجرا نمی‌شوند و فقط برای درک بهتر برنامه به کار می‌روند.



شکل ۳۴- استفاده از آکولادها برای مشخص کردن بخش‌های مختلف برنامه

- برای توضیح یک خطی از علامت // استفاده می‌شود. مثلاً:
LED را روشن می‌کند //
- برای توضیحات چند خطی از /* ... */ استفاده می‌شود.

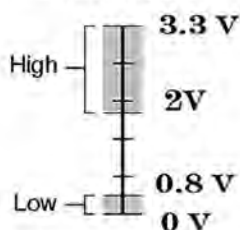
درباره اهمیت نوشتن توضیحات (Comment) در کدها و تأثیر آن بر خوانایی، نگهداری و همکاری گروهی در پروژه‌های واقعی تحقیق کنید و نتایج را در کلاس ارائه دهید.

پژوهش



خروجی دیجیتال و کنترل LED

سطوح ولتاژ در ESP32



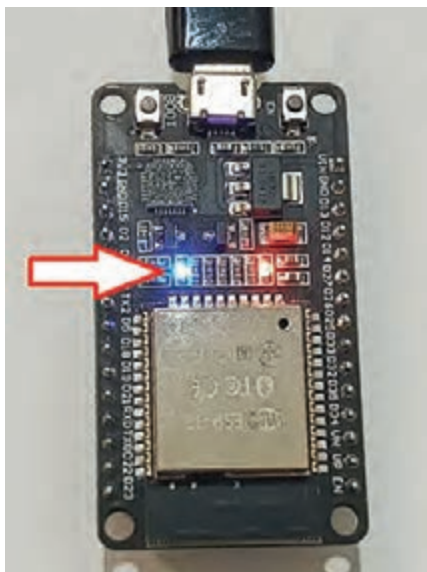
خروجی دیجیتال به پایه‌هایی از میکروکنترلر گفته می‌شود که می‌توانند تنها در دو وضعیت مشخص صفر منطقی (Low) و یک منطقی (High) قرار بگیرند. در حالت High، پایه خروجی ولتاژی برابر با ۳٫۳ ولت ایجاد می‌کند و در حالت Low ولتاژ آن صفر ولت است. (شکل ۳۵) با استفاده از خروجی‌های دیجیتال می‌توان قطعات مختلفی مانند LED، رله، بازو و عملگرهای دیگر را کنترل کرد.

شکل ۳۵- سطوح ولتاژ منطقی صفر و یک در ESP32

در ادامه، با استفاده از یک برنامه ساده، نحوه کنترل LED توسط میکروکنترلر ESP32 بررسی می‌شود.

■ مدار چشمک‌زن با کمک بُرد توسعه ESP32

در بسیاری از بُردهای توسعه ESP32، یک LED داخلی بر روی بُرد قرار دارد که معمولاً به پایه GPIO 2 متصل است. (شکل ۳۶)



در این قسمت از درس، از همین LED داخلی استفاده می‌شود و نیازی به اتصال LED خارجی وجود ندارد. برای انجام این پروژه، نیازی به اتصال مدار خارجی وجود ندارد و تنها کافی است بُرد ESP32 با استفاده از کابل USB به کامپیوتر متصل شده و تنظیمات اولیه انجام شود.

در برنامه چشمک‌زن ابتدا باید پایه مورد نظر که در اینجا پایه شماره ۲ است را خروجی در نظر بگیریم. برای این کار باید در بخش `setup()` این پایه را خروجی تعریف کنیم. در حلقه `loop()` هم باید دائماً این پایه را صفر و یک منطقی نماییم تا LED متصل شده به آن روشن و خاموش شود.

شکل ۳۶- LED آبی رنگ متصل به پایه شماره ۲ در ESP32

کد برنامه:

```
void setup() {  
    pinMode (2, OUTPUT); // پین شماره ۲ را به عنوان خروجی تنظیم می‌کند  
}  
void loop() {  
    digitalWrite (2, HIGH); // LED روشن می‌شود → ولتاژ ۳/۳ ولت به پین ۲ اعمال می‌شود  
    delay(1000); // ۱۰۰۰ میلی‌ثانیه (۱ ثانیه) صبر می‌کند  
    digitalWrite(2, LOW); // LED → ولتاژ صفر ولت (زمین) به پین ۲ اعمال می‌شود  
    خاموش می‌شود  
    delay(1000); // دوباره ۱ ثانیه صبر می‌کند  
}
```

در جدول ۸ عملکرد دستورهای مورد استفاده در برنامه شرح داده شده است.

جدول ۸- دستورات مورد استفاده در برنامه چشمک زن

مثال	پارامترها	عملکرد	دستور
pinMode(2, OUTPUT); پایه شماره ۲ خروجی می‌شود	شماره پایه: Pin: حالت پایه: Mode: output: برای خروجی input: برای ورودی	این دستور یک پایه را به عنوان ورودی یا خروجی تنظیم می‌کند.	pinMode (Pin, Mode);
digitalWrite(2, HIGH); بعد از این دستور ولتاژ پایه ۲ برابر با یک منطقی یا ۳٫۳ ولت خواهد شد	شماره پایه: Pin: وضعیت منطقی پایه: Value: HIGH معادل یک منطقی یا ۳٫۳ ولت و LOW معادل صفر منطقی یا صفر ولت	این دستور مقدار دیجیتال HIGH یا LOW را به یک پایه خروجی اعمال می‌کند.	digitalWrite (Pin, Value);
delay(1000); به مدت ۱۰۰۰ میلی ثانیه یا ۱ ثانیه اجرای برنامه متوقف شود	مدت زمان توقف برنامه: time: زمان بر حسب میلی ثانیه	این دستور اجرای برنامه را به مدت زمان مشخصی (بر حسب میلی ثانیه) متوقف می‌کند.	delay (time);

در این برنامه ابتدا در بخش تنظیمات پایه شماره ۲ را به عنوان خروجی تنظیم می‌کنیم. سپس در بخش loop ابتدا مقدار پایه شماره ۲ را یک منطقی کرده تا LED روشن شود، سپس ۱۰۰۰ میلی ثانیه توقف می‌کند و بعد LED خاموش می‌شود و ۱۰۰۰ میلی ثانیه خاموش باقی می‌ماند و این روشن و خاموش شدن دائماً تکرار می‌شود.

همان گونه که گفته شد زبان برنامه نویسی C به حروف کوچک و بزرگ حساس است بنابراین توجه داشته باشید که در استفاده از این سه دستور حتماً به کوچک و بزرگ بودن حروف دقت داشته باشید، مثلاً در دستور pinMode فقط حرف M با حرف بزرگ نوشته می‌شود.

نکته



ساخت مدار چشمک زن با کمک بُرد توسعه ESP32

هدف: آشنایی با نحوه کنترل خروجی دیجیتال میکروکنترلر ESP32 و مشاهده عملکرد آن از طریق روشن و خاموش شدن LED روی بُرد توسعه

مواد، ابزار و تجهیزات: رایانه، نرم افزار Arduino IDE، بُرد توسعه ESP32، کابل USB مناسب (برای بُردهای قدیمی تر از کابل micro USB و برای بُردهای جدیدتر از کابل type C استفاده می‌شود).

رایانه یا لپ تاپ، نرم افزار Arduino IDE، بُرد توسعه ESP32، کابل USB مناسب بُرد، اینترنت جهت دسترسی به وبسایت شبیه سازی Wokwi

مراحل اجرای کار:

۱ بعد از نوشتن برنامه و اتصال بُرد ESP32 به سیستم با کابل USB مناسب، ابتدا از منوی Tools بُرد را انتخاب کنید. (مانند شکل ۳۰)

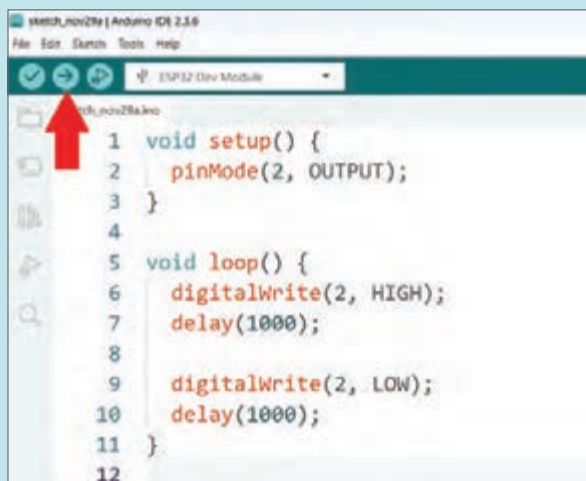
۲ از همان منوی Tools پورت را مشخص کنید.

کار عملی ۲



۳ با زدن دکمه Upload (→) (مانند شکل ۳۷) یا فشار دادن کلیدهای Ctrl+U برنامه را روی بُرد آپلود کنید.

با زدن دکمه آپلود برنامه ابتدا از نظر صحت دستورات بررسی می‌شود، سپس عمل ترجمه (کامپایل) انجام می‌شود و در آخر روی بُرد بارگذاری (آپلود) می‌شود.



شکل ۳۷- آپلود کردن برنامه

در برخی بُردها در حین انجام عمل آپلود برنامه باید دکمه BOOT را فشرده نگه داشت. (شکل ۳۸)



شکل ۳۸- دکمه Boot در برخی بُردها هنگام آپلود برنامه باید نگه داشته شود.

نکته مهم



۴ بعد از تکمیل عملیات آپلود LED آبی رنگ روی بُرد شروع به روشن و خاموش شدن می‌نماید.
۵ برنامه را طوری تغییر دهید که LED به مدت دو ثانیه روشن و یک ثانیه خاموش شود و این عمل تکرار شود.

در تمام پروژه‌های این واحد یادگیری، بُرد ESP32 نیازی به مدار تغذیه جداگانه ندارد و با اتصال کابل USB به کامپیوتر یا آداپتور مناسب روشن و آماده کار می‌شود.

نکته مهم



شبیه‌سازی پروژه‌های ESP32

برخلاف بُردهای قدیمی‌تر مانند AVR، نرم‌افزارهای رایج شبیه‌سازی مانند Proteus فاقد کتابخانه رسمی و پایدار برای ESP32 هستند. بهترین راه‌حل برای شبیه‌سازی مطمئن و سریع، استفاده از سایت wokwi.com است. این سایت یک شبیه‌ساز کاملاً آنلاین است که از میکروکنترلر ESP32 و برخی دیگر از میکروکنترلرهای جدید پشتیبانی می‌کند.

■ مزایای استفاده از Wokwi:

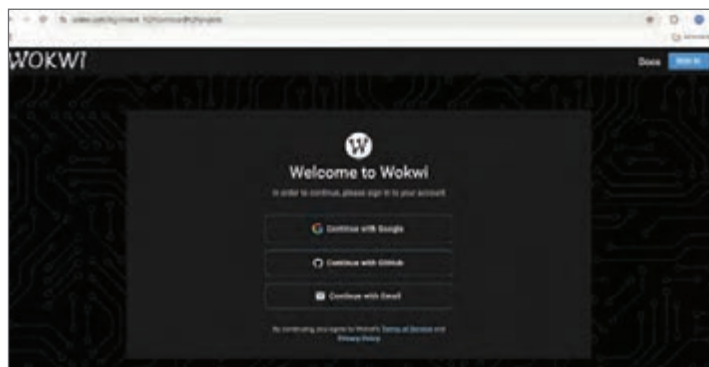
- نیازی به نصب نرم‌افزار نیست، فقط یک مرورگر وب مثل Chrome یا Firefox کافی است.
 - کدنویسی و اجرا در یک محیط انجام می‌شود. کد برنامه را مستقیماً در سایت نوشته و با یک کلیک می‌توان آن را اجرا کرد، بدون آنکه به Arduino IDE، درایور و کابل USB نیاز باشد.
 - شبیه‌سازی را علاوه بر رایانه شخصی و لپ‌تاپ، می‌توان با گوشی هوشمند و تبلت نیز انجام داد. فقط کافیست به مرورگر و اینترنت دسترسی داشت.
- استفاده از شبیه‌ساز نرم‌افزاری فرصت یکسان برای همه کسانی که به بُرد فیزیکی یا حتی رایانه دسترسی ندارند فراهم می‌کند که بتوانند بدون محدودیت، پروژه‌ها را شبیه‌سازی کرده و با تمرین مهارت خود را تقویت نمایند.

نکته



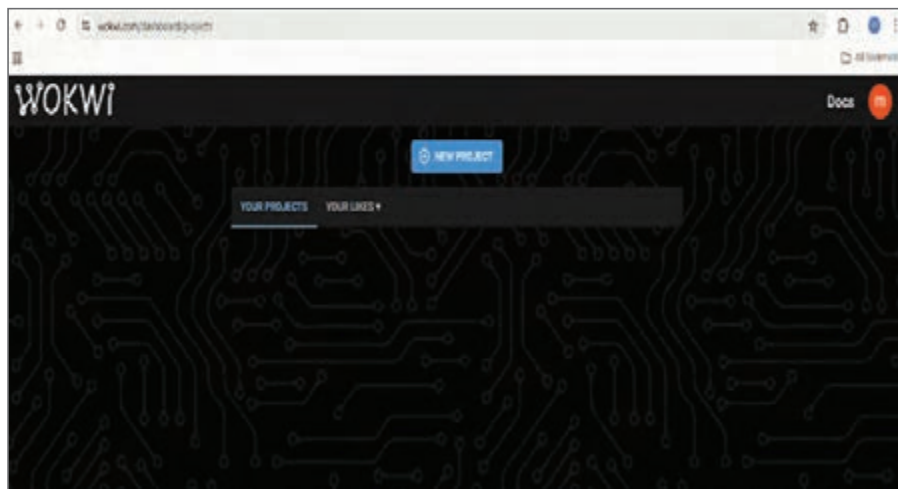
شبیه‌ساز Wokwi جایگزین کار عملی در کارگاه نیست، بلکه ابزاری قدرتمند برای یادگیری پیش از کار عملی و تثبیت مفاهیم پس از کلاس است. استفاده از آن با صلاحدید هنرآموزان محترم و امکانات موجود در هنرستان می‌باشد و می‌توان قبل از انجام کارهای عملی، به کمک شبیه‌ساز عملکرد برنامه نوشته شده را بررسی کرد و یا در صورت کمبود فرصت در منزل این کار را انجام داد.

برای انجام شبیه‌سازی در این محیط کافی است در مرورگر خود آدرس سایت wokwi.com/projects را وارد کنید. وقتی برای اولین بار وارد سایت شوید صفحه‌ای مانند شکل ۳۹ ظاهر می‌شود که از شما می‌خواهد با اکانت گوگل یا دیگر اکانت‌هایی که اشاره شده وارد شوید. بعد از ورود از طریق هر یک از گزینه‌های پیشنهادی وارد صفحه‌ای مانند شکل ۴۰ می‌شوید.



شکل ۳۹- اولین ورود به سایت شبیه ساز

در صورتی که امکان ورود به حساب کاربری در وبسایت Wokwi فراهم نبود، می‌توانید بدون ورود به حساب کاربری، از همان صفحه اصلی سایت wokwi.com، بخش پروژه‌های پرتعداد (Popular Projects) را انتخاب کرده و نمونه مدارها و برنامه‌ها را مشاهده کنید و مطابق مدار و پروژه خود تغییرات لازم را ایجاد کنید. در این صفحه پروژه‌هایی که جزء علاقه‌مندی‌هایتان است یا پروژه‌هایی که قبلاً ذخیره کرده‌اید، نشان داده می‌شود.



شکل ۴۰- محیط سایت بعد از ورود

در اولین ورود به سایت، پروژه‌ای مشاهده نخواهید کرد. برای ساخت اولین پروژه کافی است بر روی کلید NEW PROJECT کلیک کنید. با کلیک روی این گزینه صفحه‌ای مانند شکل ۴۱ باز می‌شود که از شما می‌خواهد بُرد یا میکرو مورد نظرتان را برای شبیه‌سازی انتخاب کنید. همان‌گونه که مشاهده می‌کنید در سمت چپ نام بُردهای توسعه مختلف از قبیل ESP32، STM32، PICO و... نشان داده شده که می‌توان در این سایت شبیه‌سازی کرد. همچنین انواع مختلف هر یک از این بُردها قابل شبیه‌سازی هستند.



شکل ۴۱- ساخت اولین پروژه



شکل ۴۲- انتخاب IDE مورد نظر برای شبیه‌سازی

برای شبیه‌سازی بُرد ESP32 Wroom می‌توان اولین گزینه که با نام ESP32 مشخص است را انتخاب کرد که صفحه‌ای مانند شکل ۴۲ باز می‌شود و زبان برنامه‌نویسی یا IDE مورد نظرتان را باید انتخاب کنید.

در اینجا سه انتخاب وجود دارد ARDUINO، MICROPYTHON و ESP-IDF.

برای انجام پروژه‌های این کتاب گزینه ARDUINO را انتخاب کنید. بعد از این کار صفحه‌ای مانند شکل ۴۳ باز خواهد شد.

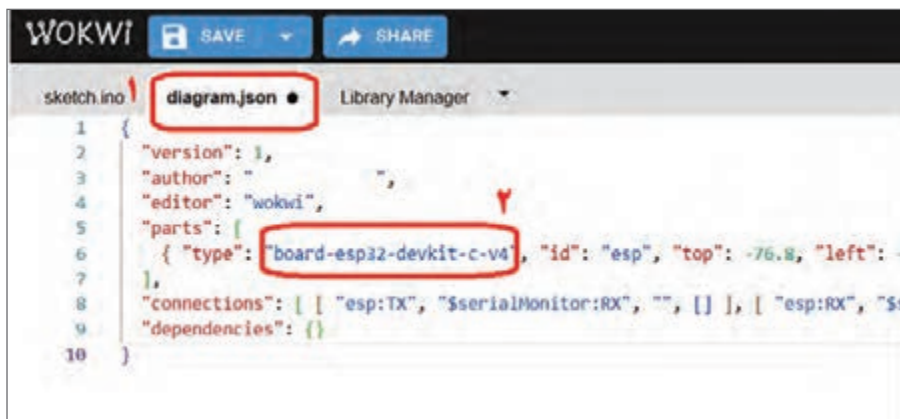


شکل ۴۳- محیط شبیه‌سازی

همان‌گونه که مشاهده می‌شود این محیط دو بخش دارد. در سمت راست صفحه می‌توانید مدار را ببینید و در سمت چپ صفحه کد را بنویسید و دیگر حتی نیاز به استفاده از Arduino IDE برای نوشتن برنامه‌ها نخواهد بود.

نکته مهمی که باید به آن توجه داشت این است که ESP32 نشان داده شده در این بخش دارای ۳۸ پایه است. می‌توان از همین بُرد استفاده کرد و تفاوتی در نوشتن کد وجود ندارد، ولی در صورتی که بخواهید از بُرد ۳۰ پایه استفاده کنید، در سمت چپ زبانه diagram.json را انتخاب کنید و مانند شکل ۴۴ در بخشی که بُرد ESP32 تعریف شده است به جای devkit-c-v4 عبارت devkit-v1 را جایگزین کنید.

پودمان چهارم: انتخاب میکروکنترلر (Microcontroller) و برنامه‌ریزی آن



شکل ۴۴- تغییر نوع ESP32 به برد ۳۰ پایه

برای اینکه قطعات دیگر را به مدار اضافه کنیم در قسمت سمت راست صفحه بر روی علامت  کلیک کرده و در لیست نمایش داده شده قطعه مورد نظرتان را اضافه کنید. در شکل ۴۵ مدار چشمک زن کار عملی شماره ۲ با اتصال LED به پایه شماره ۱۳ و اتصال مقاومت به آن شبیه‌سازی شده است. برای انجام عمل شبیه‌سازی بعد از بستن مدار و نوشتن برنامه روی علامت  کلیک می‌کنیم.



شکل ۴۵- شبیه‌سازی مدار چشمک‌زن در سایت

جهت دسترسی به مدار شبیه‌سازی شده و کد برنامه به رمزینه سریع پاسخ به اول پودمان چهارم رجوع کنید.

توجه



■ تعریف متغیر و استفاده از آن در برنامه

در برنامه چشمک‌زن کار عملی قبلی، شماره پایه و زمان تأخیر به صورت عدد مستقیم در برنامه نوشته شده‌اند. به عنوان مثال در دستور `pinMode(2, OUTPUT);` شماره پایه و در دستور `delay(1000);` زمان تأخیر به طور مستقیم نوشته شده است.

در برنامه‌نویسی، به این نوع اعداد ثابت که مستقیماً در کد نوشته می‌شوند، عدد جادویی (magic number) گفته می‌شود. استفاده زیاد از این اعداد باعث کاهش خوانایی برنامه و سخت‌تر شدن ویرایش آن می‌شود. برای حل این مشکل، از متغیر استفاده می‌کنیم.

■ **متغیر:** در برنامه‌نویسی، متغیر مکانی در حافظه میکروکنترلر است که برای ذخیره‌سازی داده‌ها استفاده می‌شود. هر متغیر شامل یک نام و یک نوع داده است که می‌تواند یک مقدار را در خود نگه دارد. مقدار ذخیره‌شده در متغیر می‌تواند در طول اجرای برنامه تغییر کند؛ به همین دلیل به آن «متغیر» گفته می‌شود.

■ **ساختار تعریف متغیر:** برای تعریف یک متغیر در آردوینو (که مبتنی بر زبان C/C++ است)، ابتدا نوع داده متغیر، سپس نام متغیر و در صورت نیاز مقدار اولیه آن نوشته می‌شود. فرم کلی تعریف متغیر به صورت جدول ۹ است.

جدول ۹- تعریف متغیر

مثال	تعریف	ساختار
int number; متغیر از نوع integer (عدد صحیح) و با نام number	در این حالت متغیری با نوع داده مشخص و با نام دلخواه معرفی می‌شود	نام متغیر نوع متغیر
int number=12; متغیر از نوع integer (عدد صحیح) و با نام number و مقدار اولیه ۱۲	در این حالت ضمن تعریف متغیر مقدار اولیه به آن تخصیص داده می‌شود	مقدار اولیه - نام متغیر نوع متغیر

مقداردهی اولیه اختیاری می‌باشد.

نکته



در زبان C انواع مختلفی از متغیرها وجود دارد که مهم‌ترین آنها در جدول ۱۰ معرفی شده‌اند.

جدول ۱۰- سه نوع از مهم‌ترین متغیرها در زبان C

متغیر	نوع	مثال
int	برای ذخیره اعداد صحیح	اعدادی مثل ۱۰، ۳- و ۲۵
float	برای ذخیره اعداد اعشاری	۱۴، ۳، ۲، ۵ و ۱۴، ۵-
char	برای ذخیره کاراکترها	'A'، 'b'، ...

درباره انواع متغیرها در زبان C و تفاوت‌های آنها از نظر فضایی که از حافظه اشغال می‌کنند تحقیق کنید و در کلاس ارائه دهید.

تحقیق کنید



■ دلایل استفاده از متغیر:

استفاده از متغیر باعث می‌شود برنامه خوانا تر شود، تغییر مقادیر آسان تر انجام پذیرد، از تکرار اعداد ثابت در کد جلوگیری شود و در نهایت برنامه‌نویسی ساخت یافته‌تری داشته باشیم.

نکته



در زبان C و آردوینو نام متغیرها نباید با عدد شروع شود و نباید شامل فاصله باشد. همچنین نام متغیر به حروف کوچک و بزرگ حساس است، مثلاً متغیر LEDpin و متغیر LEDPin دو متغیر با نام‌های متفاوت هستند.

■ برنامه چشمک‌زن با LED با استفاده از متغیر:

این برنامه همراه با توضیحات مربوط به هر سطر آن به صورت ذیل می‌باشد:

int LEDPin = 2;	شماره پایه‌ای که خروجی در نظر گرفتیم
int delayTime = 1000;	زمان تأخیر (به میلی‌ثانیه)
void setup() {	بلوک مربوط به تنظیمات setup
pinMode(LEDPin, OUTPUT);	پایه ۲ را به عنوان خروجی تعریف کردیم
}	پایان بلوک setup
void loop() {	شروع بلوک loop
digitalWrite(LEDPin, HIGH);	پایه شماره ۲ یک منطقی می‌شود و LED متصل به آن روشن می‌شود
delay(delayTime);	به اندازه ۱۰۰۰ میلی ثانیه برنامه متوقف شده و LED روشن می‌ماند
digitalWrite(LEDPin, LOW);	پایه شماره ۲ صفر منطقی می‌شود و LED متصل به آن خاموش می‌شود
delay(delayTime);	به اندازه ۱۰۰۰ میلی ثانیه LED خاموش می‌ماند
}	پایان بلوک loop

نکته



استفاده از متغیر، تنها یک «تکنیک کدنویسی» نیست، بلکه نشانه تفکر مهندسی است که شامل ساده‌سازی، انعطاف‌پذیری و آمادگی برای توسعه برنامه می‌باشد.

چشمک‌زن با LED خارجی

هدف: یادگیری اتصال LED خارجی به ESP32 و کنترل آن با برنامه.
مواد، ابزار و تجهیزات: رایانه، نرم‌افزار Arduino IDE، بُرد توسعه ESP32، کابل USB مناسب، LED (هر رنگ دلخواه)، برد برد (۲ عدد)، سیم‌های رابط. مقاومت 150Ω

کار عملی ۳



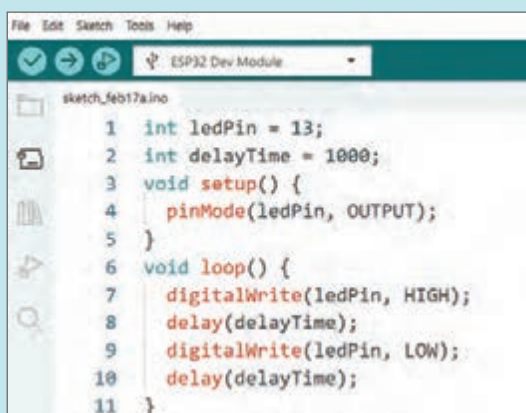
مراحل اجرای کار:

۱ ابتدا مدار را مانند شکل ۴۶ ببندید.

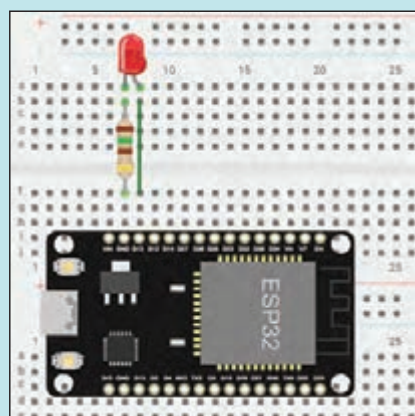
پایه آند LED را به یک پایه خروجی (مثلاً GPIO13) و پایه کاتد LED را به یک سر مقاومت وصل کنید. سر دیگر مقاومت را به زمین مدار (GND) وصل کنید.

لازم به ذکر است با توجه به ابعاد بُرد ESP32 اگر آن را روی یک بُرد قرار دهیم به جز یک ردیف بقیه سوراخ‌های بُرد زیر ESP32 قرار می‌گیرد بنابراین بهتر است از دو بُرد بُرد استفاده شود.

۲ برنامه نوشته شده در شکل ۴۷ را در محیط آردوینو بنویسید.



شکل ۴۷- برنامه مربوط به مدار چشمک زن



شکل ۴۶- نحوه سیم‌کشی مدار چشمک زن

جهت دسترسی به مدار شبیه‌سازی شده و کد برنامه به رمزینه سریع پاسخ اول پودمان چهارم رجوع کنید.

توجه



۳ بُرد ESP32 را با کابل USB مناسب به کامپیوتر متصل کنید.

۴ در منوی Tools پورت اختصاص داده شده به بُرد ESP32 را مشخص کنید.

۵ بعد از نوشتن برنامه دکمه آپلود را بزنید (از کلیدهای ترکیبی Ctrl + U نیز می‌توانید استفاده کنید).

۶ بعد از انجام عملیات آپلود چه اتفاقی می‌افتد؟ نتیجه را شرح دهید.

۷ در خط ۱۰ برنامه، دستور delay را حذف کنید. چه اتفاقی خواهد افتاد و وضعیت LED به چه صورتی خواهد شد؟

تکلیف



نکته



مدار فوق را در محیط شبیه‌سازی wokwi.com شبیه‌سازی کنید و عملکرد مدار را بررسی کنید.

ولتاژ پایه خروجی ESP32 در حالت High برابر $3/3$ ولت می‌باشد. اما یک LED معمولی بسته به نوع و رنگ آن، بین $1/8$ تا 3 ولت نیاز دارد. ولتاژ موردنیاز LED قرمز حدود $1/8$ ولت و LED آبی رنگ حدود 3 ولت است. جریان مطلوب عبوری از LED هم بین 5 تا 20 میلی‌آمپر می‌باشد. اگر LED را مستقیماً به خروجی میکروکنترلر وصل کنیم، جریان زیادی از آن عبور کرده و باعث آسیب رسیدن به LED و از آن مهمتر به ESP32 می‌شود، زیرا حداکثر جریانی که می‌توان از ESP32 کشید 40 میلی‌آمپر است. برای محدود کردن جریان کشیده شده از میکروکنترلر از مقاومت استفاده می‌کنیم. با توجه به قانون اهم، مقدار مقاومت مورد نیاز از رابطه زیر محاسبه می‌شود:

$$R = \frac{3/3V - V_f}{I_f}$$

که در آن:

V_f : ولتاژ روشن شدن LED (معمولاً برای LED قرمز: $1/8V$ ، LED زرد و سبز: $2V$ و LED آبی و سفید: $3V$)

I_f : جریان مطلوب (معمولاً 10 mA)

به‌طور مثال برای یک LED قرمز رنگ مقاومت مناسب برابر است با:

$$R = \frac{3/3V - 1/8V}{10\text{ mA}} = 150\Omega$$

می‌توان از مقاومت 150 اهمی و یا بیشتر برای اتصال LED به ESP32 استفاده کرد. هر چقدر مقاومت بیشتر باشد نور کمتر می‌شود ولی جریانی که از ESP32 کشیده می‌شود کمتر است.

■ دریافت اطلاعات ورودی دیجیتال:

در قسمت‌های قبلی از میکروکنترلر تنها برای فرمان دادن استفاده می‌شد (مانند روشن و خاموش کردن LED)، اما در بسیاری از کاربردها میکروکنترلر باید بتواند اطلاعاتی را از بیرون دریافت کند. مانند:

■ **فشرده شدن یک کلید:** مانند کلیدهای آسانسور، کلیدهای لباسشویی یا مایکروویو ...

■ **سنسور باز بودن درب یخچال یا خودرو:** که شامل یک آهنربا و سنسور (Reed Switch) می‌باشد و اگر درب باز شود، مدار قطع یا وصل می‌شود و میکروکنترلر یک سیگنال دیجیتال دریافت می‌کند تا بوق بزند یا چراغ سقف را روشن کند.



شکل ۴۸- شمارش محصولات روی نقاله

■ شمارش محصولات روی نوار نقاله (شکل ۴۸): با

عبور محصول از مقابل سنسور، نور قطع شده و توسط یک سنسور نوری (چشمی) یک پالس دیجیتال به میکروکنترلر ارسال می‌شود. میکروکنترلر با شمردن این پالس‌ها، تعداد محصول تولید شده را می‌شمارد.

در همه موارد گفته شده، اطلاعات به صورت سیگنال دیجیتال به شرح زیر به میکروکنترلر وارد می‌شود:

صفر منطقی (LOW) به معنای عدم وجود سیگنال یا اتصال به زمین
یک منطقی (HIGH) به معنای وجود سیگنال یا اتصال به ولتاژ (۳/۳ ولت)
برای خواندن وضعیت یک پایه، دو مرحله لازم است:

۱ تنظیم پایه به عنوان ورودی در تابع `setup()`:

```
pinMode (INPUT, شماره پایه); // ورودی ساده
pinMode (INPUT_PULLUP, شماره پایه); // ورودی با مقاومت بالاکش داخلی
```

۲ خواندن مقدار پایه در تابع `loop()`:

```
int x = digitalRead(4); // مقدار پایه ۴ را بخوان و در متغیر x ذخیره کن
// مقدار digitalRead همیشه یکی از دو مقدار HIGH یا LOW (صفر یا یک) است.
```

کار عملی ۴



کنترل LED با سوئیچ

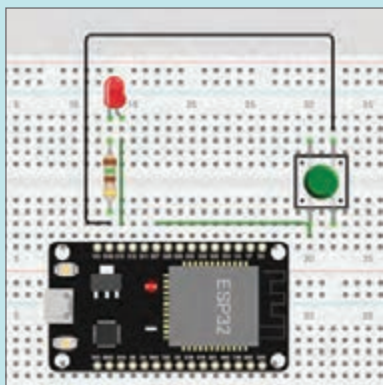
هدف: آشنایی با خواندن ورودی دیجیتال و انتقال مستقیم آن به خروجی - بدون استفاده از دستور شرطی.

مواد، ابزار و تجهیزات: بُرد توسعه ESP32 (۱ عدد)، LED (۱ عدد)، مقاومت 150Ω (۱ عدد)، کابل USB مناسب، سوئیچ فشاری (Push Button) (۱ عدد)، بُرد برد (۲ عدد)، سیم‌های رابط

مراحل اجرای کار:

۱ مدار را مانند شکل ۴۹ بر روی بُرد بُرد ببندید.

یکی از پایه‌های سوئیچ را به زمین و پایه روبرو (قطری) آن را به پایه ۱۳ بُرد ESP32 متصل کنید.



شکل ۴۹- نحوه سیم‌کشی مدار کنترل LED با سوئیچ

۲ کد نوشته شده در شکل ۵۰ را در Arduino IDE وارد کنید.

```
1 int ledPin = 13;    // LED پین
2 int buttonPin = 27; // پین دکمه
3
4 void setup() {
5   pinMode(ledPin, OUTPUT);
6   pinMode(buttonPin, INPUT_PULLUP); // Pull-up داخلی فعال شود
7 }
8
9 void loop() {
10  int state = digitalRead(buttonPin); // خواندن وضعیت دکمه
11  digitalWrite(ledPin, !state);      // معکوس وضعیت دکمه به LED اعمال شود
12 }
```

شکل ۵۰- برنامه مربوط به کنترل LED با سوئیچ

■ توضیح برخی از دستورات:

در خط ۶ پایه ۲۷ به عنوان ورودی انتخاب شده و مقاومت داخلی بالا کش (Pull Up) فعال می‌شود. یعنی در حالت معمول و بدون ورودی مقدار این پایه HIGH یا یک منطقی است. در خط ۱۰ متغیر state از نوع عدد صحیح تعریف شده که مقدار پایه ورودی (پایه ۲۷) را داخل آن قرار می‌دهیم و در خط ۱۱، با استفاده از علامت ! (معکوس منطقی)، مقدار متغیر state برعکس شده و به پایه LED ارسال می‌شود. به این ترتیب، هنگامی که دکمه فشرده می‌شود (پایه ورودی LOW می‌شود)، LED روشن شده و هنگامی که دکمه رها می‌شود (پایه ورودی HIGH می‌شود)، LED خاموش می‌گردد.

جهت دسترسی به مدار شبیه‌سازی شده و کد برنامه به رمزینه سریع پاسخ اول پودمان چهارم رجوع کنید.

توجه



۳ بُرد ESP32 را با کابل USB مناسب به کامپیوتر متصل کنید.

۴ در منوی Tools، پورت اختصاص داده شده به بُرد ESP32 را مشخص کنید.

۵ دکمه آپلود را بزنید (یا کلیدهای ترکیبی Ctrl + U).

۶ بعد از انجام عملیات آپلود چه اتفاقی می‌افتد؟ نتیجه را شرح دهید.

۷ در صورتی که در برنامه از علامت ! استفاده نشود، با فشردن دکمه چه اتفاقی می‌افتد؟ شرح دهید.

۸ پایه سوئیچ را به جای زمین به تغذیه (۳/۳V) وصل کنید، حال چه تغییری باید در برنامه ایجاد شود تا عملکرد مدار تغییر نکند؟

اگر بخواهید پایه سوئیچ را به جای زمین به تغذیه وصل کنید، باید توجه شود که به ولتاژ ۵ ولت وصل نشود، زیرا ولتاژ ورودی مجاز پایه‌های ESP32 فقط ۳/۳ ولت است و با اتصال مستقیم ولتاژ ۵ ولت یا بیشتر به پایه‌های ورودی ESP32، بُرد آسیب می‌بیند.

نکته ایمنی



ساختارهای تصمیم یا ساختارهای شرطی

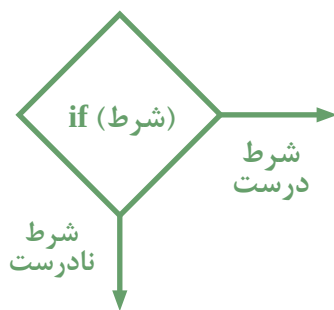
در برنامه‌هایی که تا اینجا نوشته شد، دستورات به صورت خطی و به ترتیب اجرا می‌شدند. اما در کاربردهای واقعی، میکروکنترلرها باید بتوانند بر اساس شرایط مختلف تصمیم‌گیری کنند، برای مثال، یک دستگاه ممکن است تنها در صورتی یک خروجی را فعال کند که دکمه‌ای فشرده شده باشد یا یک شرط خاص برقرار باشد. برای پیاده‌سازی چنین رفتارهایی، از ساختارهای تصمیم‌گیری یا شرطی در برنامه‌نویسی استفاده می‌شود. به عنوان مثال:

■ اگر دکمه فشرده شد، شروع به شمارش کند.

■ اگر دما از ۳۰ درجه بیشتر شد، فن به کار بیفتد.

■ اگر درب باز بود، هشدار صوتی فعال شود.

این نوع تصمیم‌گیری‌ها با استفاده از ساختارهای شرطی در برنامه‌نویسی پیاده‌سازی می‌شوند. در ادامه دو ساختار شرطی `if ... else` و `if` توضیح داده می‌شوند.



شکل ۵۱- ساختار شرطی `if`

■ ساختار شرطی `if`

ساختار شرطی `if` یا «اگر» برای اجرای یک یا چند دستور، تنها در صورتی استفاده می‌شود که یک شرط مشخص برقرار باشد. اگر شرط برقرار باشد، دستورات داخل `if` اجرا می‌شوند و در غیر این صورت از آنها عبور می‌شود. (شکل ۵۱) به بیان ساده، `if` به میکروکنترلر می‌گوید: اگر این شرط درست بود، این کار را انجام بده.

در جدول ۱۱ ساختار کلی دستور `if` به همراه مثال نشان داده شده است.

جدول ۱۱- ساختار کلی دستور `if`

مثال	توضیح	شکل ساختار
<pre>int x = 5; if (x > ۳) { digitalWrite(۱۳, HIGH); }</pre> چون $X = ۵$ است و بزرگ‌تر از ۳ است پس شرط برقرار بوده و پایه ۱۳ یک منطقی می‌شود	اگر شرط داخل پرانتز برقرار بود دستورات بین آکولادها اجرا می‌شوند، در غیر اینصورت این دستورات اجرا نمی‌شوند	<pre>if (شرط) { دستورها }</pre>

■ استفاده از `if` برای بررسی ورودی دیجیتال: در پروژه‌های عملی، معمولاً شرط‌ها بر اساس وضعیت ورودی‌ها نوشته می‌شوند، برای مثال، می‌توان بررسی کرد که آیا یک دکمه فشرده شده است یا خیر.

```
if (buttonState == HIGH) {
    digitalWrite (LEDPin, HIGH);}

```

نکته



در شرط‌ها معمولاً از عملگرهای مقایسه‌ای مانند جدول ۱۲ استفاده می‌شود.

جدول ۱۲- عملگرهای مقایسه‌ای مهم مورد استفاده در ساختارهای شرطی

عملگر	مثال	توضیح
تساوی: ==	$A == 5$	بررسی می‌کند که متغیر A برابر با ۵ است یا نه
بزرگ‌تر: >	$A > 5$	بررسی می‌کند متغیر A بزرگ‌تر از ۵ است یا نه
کوچک‌تر: <	$A < 8$	بررسی می‌کند متغیر A کوچک‌تر از ۸ است یا نه
مخالف: !=	$A != 10$	بررسی می‌کند متغیر A مخالف ۱۰ است یا نه

نکته



علامت‌های = و == در برنامه‌نویسی کاملاً با یکدیگر متفاوت هستند و نباید به جای هم استفاده شوند. اشتباه گرفتن این دو علامت، یکی از خطاهای متداول در نوشتن برنامه است. در جدول ۱۳ تفاوت این دو علامت با مثال نشان داده شده است.

جدول ۱۳- تفاوت عملگر تساوی و انتساب (جایگزینی)

علامت	مثال	توضیح
انتساب: =	$Num = 5;$	بعد از این دستور مقدار متغیر Num برابر با عدد ۵ می‌شود
تساوی: ==	$Num == 8$	مقدار متغیر Num تغییر نمی‌کند، فقط با عدد ۸ مقایسه می‌شود

نکته



استفاده از علامت = به جای == در شرط‌ها باعث بروز اجرای نادرست برنامه می‌شود و در هر صورت شرط اجرا می‌شود. توجه شود در این حال کامپایلر از کد ایراد نمی‌گیرد ولی برنامه عملکرد درستی ندارد. بنابراین در نوشتن شرط‌ها باید دقت شود که برای مقایسه، حتماً از == استفاده گردد.

■ ساختار شرطی if ... else

در برخی برنامه‌ها، تنها اجرای یک دستور در صورت برقرار بودن شرط کافی نیست و لازم است در صورت برقرار نبودن شرط نیز یک عمل مشخص انجام شود. برای این منظور از ساختار شرطی if ... else استفاده می‌شود. به بیان ساده، if ... else به میکروکنترلر می‌گوید: «اگر شرط برقرار بود، این کار را انجام بده و در غیر این صورت، کار دیگری انجام بده»

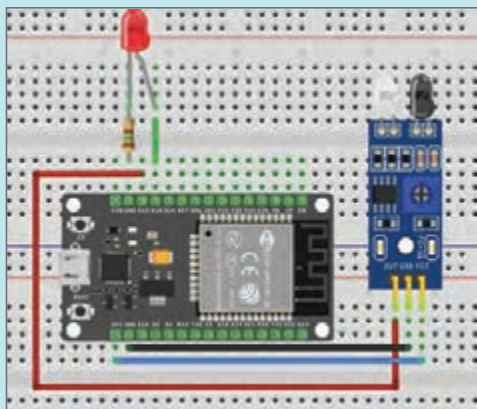
```

if (شرط) {
    دستورها در صورت درست بودن شرط
}
else {
    دستورها در صورت نادرست بودن شرط
}
    
```




تشخیص عبور قطعه در خط تولید با سنسور مادون قرمز (IR)

هدف: به کارگیری دستور شرطی if در یک مثال کاربردی و شبیه سازی تشخیص عبور قطعه در خط تولید با استفاده از سنسور مادون قرمز



شکل ۵۲- مدار تشخیص عبور قطعه

مواد، ابزار و تجهیزات: بُرد توسعه ESP32 (۱ عدد)، LED (۱ عدد)، مقاومت 150Ω (۱ عدد)، ماژول سنسور مادون قرمز IR Sensor مدل HW201 (۱ عدد)، بُرد بُرد (۲ عدد)، سیم های رابط، کابل USB مناسب.

مراحل اجرای کار:

۱ مانند شکل ۵۲ پایه Vcc ماژول سنسور مادون قرمز را به پایه 3v3 بُرد ESP32 و پایه GND ماژول را به یکی از پایه های GND بُرد ESP32 متصل کنید، پایه OUT ماژول را هم به پایه ۱۳ بُرد ESP32 متصل کنید، سپس یک پایه مقاومت 150Ω اهمی را به GND بُرد ESP32 و پایه بعدی را به کاتد LED متصل کنید. پایه آند LED را هم به پایه شماره ۱۲ بُرد ESP32 وصل کنید.

۲ برنامه شکل ۵۳ را در محیط آردوینو بنویسید.

۳ بُرد ESP32 را با کابل USB مناسب به کامپیوتر متصل کنید.

۴ در منوی Tools پورت اختصاص داده شده به بُرد ESP32 را مشخص کنید.

۵ دکمه آپلود را بزنید. (یا کلیدهای ترکیبی Ctrl+U)

```

sketch_feb17a.ino
1  int ledPin = 12;
2  int IrPin=13;
3  int delayTime =1000;
4  int number;
5  void setup() {
6      pinMode(ledPin, OUTPUT);
7      pinMode(IrPin, INPUT_PULLUP);
8  }
9  void loop() {
10
11      if(digitalRead(IrPin)==0)
12      {
13          number++;
14          delay(100);
15      }
16
17      if(number==6)
18      {
19          digitalWrite(ledPin,HIGH);
20          delay(delayTime);
21          number=0;
22          digitalWrite(ledPin,LOW);
23      }
24  }
    
```

شکل ۵۳- برنامه مربوط به شبیه سازی تشخیص عبور قطعه

۶ بعد از آپلود شدن برنامه روی بُرد، اگر قطعه‌ای در محدوده‌ای تا حدود ۱۵ سانتی‌متر از جلوی ماژول عبور کند شمارش انجام خواهد شد و بعد از عبور ۶ عدد قطعه از جلوی سنسور، LED متصل به پایه شماره ۱۲ روشن می‌شود.

۷ عملکرد مدار را شرح دهید.

جهت دسترسی به مدار شبیه‌سازی شده و کد برنامه به رمزیننه سریع پاسخ اول پودمان چهارم رجوع کنید.

توجه



نکته



۱ علت قرار دادن دستور delay(100) در خط ۱۴ این است که در هنگام عبور یک قطعه از مقابل سنسور، چند بار شمارش انجام نشود. بر اساس سرعت عبور قطعه این زمان تأخیر را می‌توان تغییر داد.

۲ پیچ آبی روی ماژول سنسور مادون قرمز برای تنظیم حساسیت ماژول است که می‌توان فاصله تشخیص مانع را بین ۵ تا ۳۰ سانتی‌متر تغییر داد.

۳ در صورتی که ماژول فرستنده و گیرنده مادون قرمز در دسترس نبود می‌توان از دکمه فشاری (Push Button) برای شبیه‌سازی عملکرد استفاده نمود.

توجه شود که پایه VCC ماژول فرستنده و گیرنده مادون قرمز را نباید به ولتاژ ۵ ولت متصل کرد، بلکه برای تغذیه این ماژول از پایه ۳/۳V میکروکنترلر استفاده شود، زیرا در صورت استفاده از ولتاژ ۵ ولت، ولتاژ پایه خروجی ماژول که به ESP32 متصل است ۵ ولت خواهد شد و باعث آسیب دیدگی بُرد ESP32 می‌شود.

نکته



سؤال ۱: در صورتی که بخواهیم به جای یک LED یک لامپ یا موتور روشن شود چه تغییری در مدار باید انجام دهیم؟



شکل ۵۴- پروژه ساخت پارکینگ هوشمند

فرض کنید یک پارکینگ با ظرفیت مشخص داریم و دارای یک درب ورودی و یک درب خروجی است. (شکل ۵۴)

می‌خواهیم مداری طراحی کنیم که هرگاه ظرفیت پارکینگ تکمیل شد، چراغ قرمز روشن شود و وقتی ظرفیت خالی ایجاد شد، چراغ سبز روشن شود. چه تغییراتی در برنامه (کد) و مدار لازم است؟

تحقیق کنید



ساختارهای تکرار

در قسمت‌های قبل دیدیم که می‌توان با نوشتن چند دستور ساده، یک LED را روشن و خاموش کرد، یا با استفاده از شرط‌ها تصمیم گرفت که چه کاری انجام شود. اما در بسیاری از کارهای واقعی، انجام یک عمل تنها برای یک بار کافی نیست و لازم است یک دستور چندین بار پشت سر هم اجرا شود. به عنوان مثال:

- وقتی می‌خواهیم یک شمارنده بسازیم که تعداد معینی شمارش انجام دهد.
- وقتی قرار است چندین LED به ترتیب روشن و خاموش شوند (مثل شمارنده‌های حلقوی و جانسون).
- وقتی قرار است چراغ‌های یک تابلوی تبلیغاتی یا ویتترین مغازه یکی یکی روشن و خاموش شوند و این کار بارها تکرار شود.

در چنین شرایطی، نمی‌توان هر بار دستور را به صورت جداگانه و دستی نوشت، زیرا باعث طولانی شدن برنامه و سخت شدن تغییر آن می‌شود. به همین دلیل، در برنامه‌نویسی از ساختارهای تکرار استفاده می‌شود که امکان اجرای منظم و کنترل‌شده یک یا چند دستور را فراهم می‌کند.

■ پارامترهای اصلی در ساختارهای تکرار:

در حلقه‌ها معمولاً سه مؤلفه کلیدی وجود دارد:

- ۱ مقدار اولیه شمارنده (شروع شمارش): مثلاً $i = 0$
 - ۲ شرط ادامه حلقه: حلقه تا زمانی ادامه می‌یابد که این شرط درست باشد، مثلاً $i < 5$
 - ۳ تغییر شمارنده در هر تکرار (گام): مثلاً $i++$ (یعنی $i = i + 1$)
- به عنوان مثال برای شمارش از ۰ تا ۴ و تکرار آن، ساختار تکرار به صورت زیر می‌باشد:
- مقدار اولیه: $i = 0$
- شرط ادامه: $i < 5$ (وقتی i برابر ۵ شد، شرط نادرست می‌شود و حلقه پایان می‌یابد)
- گام: $i++$

■ انواع ساختارهای تکرار در C (آردوینو)

در زبان C (که Arduino IDE از آن استفاده می‌کند)، سه ساختار تکرار اصلی وجود دارد:

- **for**: هنگامی که تعداد تکرارها از قبل مشخص است (مثلاً شمارش ۰ تا ۹).
 - **while**: هنگامی که تکرار تا برقرار بودن یک شرط ادامه یابد (مثلاً تا زمانی که دکمه فشرده شده است).
 - **do-while**: هنگامی که حداقل یک بار اجرای دستور ضروری باشد (مثل نمایش منوی اولیه).
- در این بخش، با حلقه **for** آشنا می‌شویم و سایر ساختارها را در پروژه‌های بعدی بررسی خواهیم کرد.

■ ساختار تکرار **for**: قالب کلی ساختار **for** به شکل زیر است:

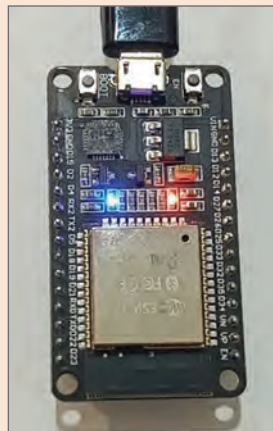
```
for (گام شمارش; شرط پایان; شروع شمارش)
{
    دستورات داخل حلقه //
}
```

مثال



فرض کنید می‌خواهیم مداری بسازیم که ۵ مرتبه LED متصل شده به پایه شماره ۲ را روشن و خاموش کند.

```
1 int ledPin = 2;
2
3 void setup() {
4   pinMode(ledPin, OUTPUT);
5   for (int i = 0; i < 5; i++) {
6     digitalWrite(ledPin, HIGH);
7     delay(500);
8     digitalWrite(ledPin, LOW);
9     delay(500);
10  }
11 }
12
13 void loop() {
14   // برنامه پس از اتمام ۵ بار چشمک‌زدن، متوقف می‌شود
15 }
```



شکل ۵۵- برنامه مداری برای روشن و خاموش شدن LED آبی رنگ برای ۵ مرتبه

در این مثال در حلقه for (خط ۵)، شروع شمارش از $i = 0$ بوده و شرط ادامه شمارش $i < 5$ و گام شمارش هم یک است. برنامه مربوط به این مدار را در شکل ۵۵ مشاهده می‌کنید.

توجه



جهت دسترسی به مدار شبیه‌سازی شده و کد برنامه به رمزینه سریع پاسخ اول پودمان چهارم رجوع کنید.

نکته



دستور $i++$ معادل $i = i + 1$ است یعنی هر بار یک عدد به عدد قبلی اضافه کن. برای کاهش ۱ واحدی هم می‌توانیم از دستور $i--$ به جای $i = i - 1$ استفاده کنیم.

کار عملی ۶



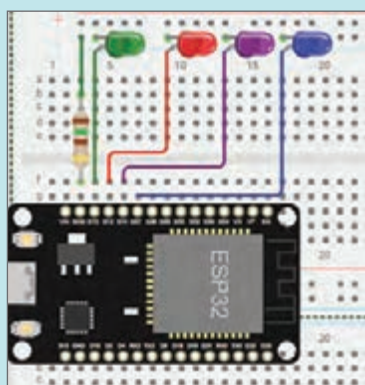
مدار رقص نور

هدف: آشنایی با ساختارهای تکرار

مواد، ابزار و تجهیزات: بُرد توسعه ESP32 (۱ عدد)، LED (۴ عدد)، مقاومت 150Ω (۱ عدد)، برد (۲ عدد)، سیم‌های رابط، کابل USB مناسب.

مراحل اجرای کار:

۱ مدار را مانند شکل ۵۶ را بر روی بُرد ببندید. پایه‌های آند LEDها را به ترتیب به پایه‌های ۱۳، ۱۲، ۱۴ و ۲۷ بُرد توسعه ESP32 و پایه‌های کاتد LEDها را به ردیف منفی بُرد متصل کنید. مقاومت 150Ω اهمی را هم بین ردیف منفی پایه GND بُرد توسعه ESP32 قرار دهید.



شکل ۵۶- نحوه سیم‌کشی مدار رقص نور

```

sketch_unov2Ba.ino
1  int led0 = 13;
2  int led1 = 12;
3  int led2 = 14;
4  int led3 = 27;
5  void setup() {
6      pinMode(led0, OUTPUT);
7      pinMode(led1, OUTPUT);
8      pinMode(led2, OUTPUT);
9      pinMode(led3, OUTPUT);
10 }
11
12 void loop() {
13     digitalWrite(led0, HIGH);
14     delay(100);
15     digitalWrite(led1, HIGH);
16     delay(100);
17     digitalWrite(led2, HIGH);
18     delay(100);
19     digitalWrite(led3, HIGH);
20     delay(100);
21     digitalWrite(led0, LOW);
22     delay(100);
23     digitalWrite(led1, LOW);
24     delay(100);
25     digitalWrite(led2, LOW);
26     delay(100);
27     digitalWrite(led3, LOW);
28     delay(100);
29 }

```

شکل ۵۷- برنامه مدار رقص نور

۲ کد نوشته شده در شکل ۵۷ را در Arduino IDE وارد کنید.

در این برنامه در خطوط ۱ تا ۴، پایه‌های GPIO بُرد ESP32 که به عنوان خروجی در نظر گرفته شده‌اند و LEDها به آنها متصل شده‌اند را در متغیرهای LED0 تا LED3 تعیین کنید. سپس در بلوک setup (خطوط ۶ تا ۹) این پایه‌ها را به عنوان خروجی تنظیم نمایید.

در بلوک loop (خطوط ۱۳ به بعد) به ترتیب و با تأخیر زمانی، این پایه‌ها را HIGH کنید تا LEDها به ترتیب روشن شوند و پس از روشن شدن آخرین LED، به ترتیب و با تأخیر زمانی آنها را LOW کنید تا LEDها خاموش شوند.

۳ بُرد ESP32 را با کابل USB مناسب به کامپیوتر متصل کنید.

۴ در منوی TOOLS پورت اختصاص داده شده به بُرد ESP32 را مشخص کنید.

۵ دکمه آپلود را بزنید. (یا کلیدهای ترکیبی Ctrl+U) و عملکرد مدار را بررسی کنید.

جهت دسترسی به مدار شبیه‌سازی شده و کد برنامه به رمزینه سریع پاسخ اول پودمان چهارم رجوع کنید.

توجه



آرایه‌ها

همان‌گونه که در کار عملی قبلی مشاهده می‌شود، برای ساخت یک رقص نور ساده با چهار LED، مجبور به تکرار چندین باره دو دستور digitalWrite و delay شدیم. اگر تعداد LEDها بیشتر شود یا بخواهیم الگوهای متنوع‌تری از رقص نور ایجاد کنیم، حجم برنامه افزایش یافته و خوانایی آن کاهش می‌یابد. در برنامه‌نویسی برای جلوگیری از تکرار بی‌مورد دستورات، از ساختارهای تکرار استفاده می‌شود. اما مشکلی که در اینجا وجود دارد این است که پایه‌های خروجی میکروکنترلر به ترتیب عددی قرار نگرفته‌اند (۱۳، ۱۲، ۱۴ و ۲۷) و بنابراین نمی‌توان آنها را به‌سادگی با یک شمارنده افزایش داد.

برای حل این مشکل، از آرایه‌ها استفاده می‌کنیم تا شماره پایه‌ها را به صورت منظم در یک مجموعه ذخیره کرده و با کمک حلقه `for` روی آنها حرکت کنیم.

■ تعریف آرایه

آرایه روشی برای نگهداری چند مقدار هم نوع در برنامه است که همگی با یک نام ذخیره می‌شوند. به عبارت ساده‌تر، آرایه مانند یک کمد چند کشو شکل ۵۸ است که هر کشو شماره مشخصی دارد و می‌تواند یک مقدار را در خود نگه دارد.

با استفاده از آرایه، به جای تعریف چند متغیر جداگانه، می‌توان چند مقدار مرتبط را به صورت منظم و ساخت یافته در یک مجموعه قرار داد.

به عنوان مثال در مدار رقص نور کار عملی قبلی، لازم است شماره چهار پایه خروجی ۱۳، ۱۲، ۱۴ و ۲۷ را به گونه‌ای ذخیره کنیم که بتوانیم به ترتیب روی آنها عملیات انجام دهیم، حتی اگر این شماره‌ها پشت سر هم نباشند. برای این کار، از یک آرایه استفاده می‌کنیم.

```
int LEDPins[4] = {13, 12, 14, 27};
```



شکل ۵۸- شماره خانه‌ها در آرایه‌ها از ۰ شروع می‌شود

■ کلمه `int` نشان می‌دهد مقادیر داخل آرایه، از نوع عدد صحیح هستند.

■ `LEDPins` نام آرایه است.

■ عدد داخل کروشه (`[4]`) تعداد خانه‌های آرایه را مشخص می‌کند.

■ مقادیر داخل آکولاد `{ }` به ترتیب در خانه‌های ۰ تا ۳ آرایه قرار می‌گیرند:

`LEDPins[0] = 13` □

`LEDPins[1] = 12` □

`LEDPins[2] = 14` □

`LEDPins[3] = 27` □

در زبان C و Arduino، شمارش خانه‌های آرایه از صفر شروع می‌شود، نه از یک.

نکته



با این روش، دیگر نیازی به تکرار دستی دستورات برای هر LED نیست؛ بلکه می‌توانیم با یک حلقه، روی تمام خانه‌های آرایه حرکت کنیم و عملیات مورد نظر را به صورت خودکار انجام دهیم.



مدار رقص نور با استفاده از آرایه

هدف: آشنایی با آرایه در برنامه‌نویسی

مواد، ابزار و تجهیزات:

برد توسعه ESP32 (۱ عدد)، LED (۴ عدد)، مقاومت $150\ \Omega$ (۱ عدد)، برد برد (۲ عدد)، سیم‌های رابط، کابل USB مناسب.

مراحل اجرای کار:

۱ مدار را مانند شکل ۵۶ (مدار کار عملی ۶) رابر روی برد برد ببندید.

۲ کد نوشته شده در شکل ۵۹ را در Arduino IDE وارد کنید.

در خط ۱، یک آرایه چهارعضوی به نام LEDPins تعریف شده و شماره پایه‌های متصل به LEDها به ترتیب در آن قرار گرفته‌اند. این کار به ما امکان می‌دهد بدون وابستگی به ترتیب عددی پایه‌ها، به صورت منطقی روی آنها عمل کنیم.

در خط ۲، متغیری به نام i به عنوان شمارنده حلقه‌ها در سطح سراسری برنامه تعریف شده است؛ یعنی این متغیر در هر دو تابع setup و loop قابل استفاده است.

خطوط ۴ تا ۷ مربوط به تابع setup هستند.

در خط ۵، حلقه‌ای تعریف شده که مقدار i را از صفر تا سه تغییر می‌دهد و در هر تکرار، با استفاده از LEDPins[i]، یکی از پایه‌ها را در خط ۵ به عنوان خروجی تنظیم می‌کند.

در تابع loop، خطوط ۱۰ تا ۱۳ یک حلقه تکرار دیگر را تشکیل می‌دهند که در هر دور، LED متناظر با اندیس i را روشن می‌کند و سپس ۱۰۰ میلی ثانیه تأخیر ایجاد می‌کند (خط ۱۲).

خطوط ۱۴ تا ۱۷ نیز حلقه‌ای مشابه را برای خاموش کردن همان LEDها، با همان ترتیب پیاده‌سازی می‌کنند. تنها تفاوت در خط ۱۵ است که به جای HIGH، مقدار LOW ارسال می‌شود. در مجموع، با تعریف یک آرایه و استفاده از دو حلقه ساده، به جای ۲۹ خط کد تکراری، تنها با ۱۸ خط، می‌توان همان رفتار را به صورت انعطاف‌پذیر و قابل گسترش پیاده‌سازی کرد.

```
1 int ledPins[4] = {13, 12, 14, 27};
2 int i;
3 void setup() {
4   for ( i = 0; i < 4; i++) {
5     pinMode(ledPins[i], OUTPUT);
6   }
7 }
8
9 void loop() {
10  for ( i = 0; i < 4; i++) {
11    digitalWrite(ledPins[i], HIGH);
12    delay(100);
13  }
14  for ( i = 0; i < 4; i++) {
15    digitalWrite(ledPins[i], LOW);
16    delay(100);
17  }
18 }
```

شکل ۵۹- برنامه مدار رقص نور به کمک آرایه

جهت دسترسی به مدار شبیه‌سازی شده و کد برنامه به رمزینه سریع پاسخ اول پودمان چهارم رجوع کنید.

توجه



۳ بُرد ESP32 را با کابل USB مناسب به کامپیوتر متصل کنید.

۴ در منوی TOOLS پورت اختصاص داده شده به بُرد ESP32 را مشخص کنید.

۵ دکمه آپلود را بزنید. (یا کلیدهای ترکیبی Ctrl+U) و عملکرد مدار را بررسی کنید.

دوباره ساختارهای تکرار در زبان C تحقیق کنید، تفاوت‌های آنها را از نظر نحوه اجرا و کاربُرد پیدا کنید و مشخص کنید هر کدام در چه شرایطی مناسب‌تر و ایمن‌تر استفاده می‌شوند.

تحقیق کنید



پروژه



در این پروژه، هدف شبیه‌سازی یک سیستم چراغ راهنمایی همراه با کنترل عبور عابر پیاده است (شکل ۶۰). به این صورت که در یک خیابان و در محل خط عابر پیاده، به منظور افزایش ایمنی عابران، چراغ‌های راهنمایی برای رانندگان و عابران پیاده نصب شده است. در حالت عادی، چراغ راهنمایی خودروها در وضعیت سبز قرار دارد و چراغ عابر پیاده قرمز است. با فشردن کلید مخصوص عابر پیاده، سیستم وارد حالت عبور عابر می‌شود؛ ابتدا چراغ خودروها به رنگ زرد تغییر می‌کند تا رانندگان برای توقف آماده شوند، سپس چراغ قرمز خودروها به صورت چشمک‌زن فعال می‌شود. پس از آن، چراغ برای خودروها قرمز و عابر پیاده سبز شده و عابران می‌توانند با ایمنی کامل از خیابان عبور کنند.



شکل ۶۰- چراغ راهنمایی قابل کنترل توسط عابر پیاده

در این پروژه، هنرجویان با استفاده از بُرد ESP32 و به‌کارگیری دستورات شرطی (if)، حلقه‌های تکرار (for)، آرایه‌ها و دریافت ورودی از سوئیچ، عملکرد یک سیستم کنترلی ساده و کاربُردی را پیاده‌سازی می‌کند.

شبیه‌سازی سیستم چراغ راهنمایی خیابان با کنترل عبور توسط عابر پیاده

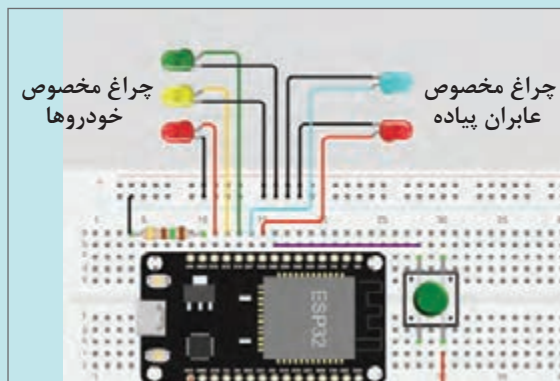
هدف: استفاده از شرط‌ها، حلقه‌ها و آرایه‌ها در قالب یک پروژه کاربُردی و نزدیک به دنیای واقعی مواد، ابزار و تجهیزات:

بُرد توسعه ESP32 (۱ عدد)، LED (۵ عدد)، مقاومت $150\ \Omega$ (۱ عدد)، بُرد برد (۲ عدد)، سیم‌های رابط، کابل USB مناسب، سوئیچ

کار عملی ۸



مراحل اجرای کار:



شکل ۶۱- سیم‌کشی مربوط به شبیه‌سازی چراغ راهنما

۱ مدار شکل ۶۱ را بر روی بردبرد ببندید. در این مدار از دو LED سبز رنگ، دو LED قرمز و یک LED زرد رنگ استفاده می‌شود، پایه‌های کاتد تمام LEDها را به ردیف زمین بردبرد متصل می‌کنیم و با یک مقاومت ۱۵۰ اهمی این ردیف را به زمین ESP32 متصل کنید.

سپس آند LED قرمز را به پایه ۱۳ برد ESP32، آند LED زرد را به پایه ۱۲ و آند LED سبز را به پایه ۱۴ برد ESP32 متصل

کنید (این سه LED برای شبیه‌سازی چراغ مخصوص خودروها استفاده شده است). در گام بعد آند LED سبز رنگ چراغ عابرین به پایه ۲۷ و آند LED قرمز رنگ به پایه ۲۶ متصل شود. یک پایه تک سوئیچ را به پایه ۳/۳ ولت ESP32 و پایه مقابل آن (به‌صورت قطری) را به پایه ۲۵ برد ESP32 متصل می‌کنیم.

توجه شود که پایه تک سوئیچ به پایه ۵ ولت ESP32 متصل نشود که باعث آسیب دیدن میکرو خواهد شد.

نکته ایمنی



```
1 // LEDهای مربوط به خودرو
2 int carLeds[3] = {13, 12, 14};
3
4 // LEDهای مربوط به عابر پیاده
5 int pedLeds[2] = {26, 27};
6
7 int buttonPin = 25;
8 void setup() {
9
10     for (int i = 0; i < 3; i++) {
11         pinMode(carLeds[i], OUTPUT);
12     }
13
14     for (int i = 0; i < 2; i++) {
15         pinMode(pedLeds[i], OUTPUT);
16     }
17
18     pinMode(buttonPin, INPUT_PULLDOWN);
19
20     digitalWrite(carLeds[2], HIGH); // سبز خودرو
21     digitalWrite(pedLeds[0], HIGH); // قرمز عابر
22 }
```

(الف)

شکل ۶۲- برنامه مدار رقص نور به کمک آرایه

۲ برنامه نوشته شده در شکل ۶۲ را در محیط Arduino IDE بنویسید.

در این برنامه در خط ۲ پایه‌های متصل به LEDهای مربوط به چراغ راهنمای خودروها را در یک آرایه ۳ عضوی به نام carLeds معرفی کردیم، که به ترتیب پایه‌های ۱۳ به LED قرمز، پایه ۱۲ به زرد و پایه ۱۴ به سبز اختصاص داده شده. در خط ۵ هم یک آرایه دیگر برای معرفی پایه‌های متصل شده به دو LED سبز و قرمز مربوط به چراغ عابر معرفی شده‌اند. در خط ۷ نیز متغیر buttonPin از نوع عدد صحیح برای معرفی پایه متصل به کلید ورودی تعریف شده است.

```

23 void loop() {
24   if (digitalRead(buttonPin) == HIGH) {
25     // ۱. سبز خودرو خاموش، زرد روشن
26     digitalWrite(carLeds[2], LOW);
27     digitalWrite(carLeds[1], HIGH);
28     delay(2000);
29     // ۲. زرد خاموش
30     digitalWrite(carLeds[1], LOW);
31     // ۳. قرمز خودرو ۴ بار چشمک بزند
32     for (int i = 0; i < 4; i++) {
33       digitalWrite(carLeds[0], HIGH);
34       delay(500);
35       digitalWrite(carLeds[0], LOW);
36       delay(500);
37     }
38     // ۴. قرمز خودرو ثابت ۹ عابر عمل
39     digitalWrite(carLeds[0], HIGH);
40     digitalWrite(pedLeds[0], LOW);
41     digitalWrite(pedLeds[1], HIGH);
42     delay(5000);
43     // ۵. بازگشت به حالت عادی
44     digitalWrite(pedLeds[1], LOW);
45     digitalWrite(pedLeds[0], HIGH);
46     digitalWrite(carLeds[0], LOW);
47     digitalWrite(carLeds[2], HIGH);
48   }
49 }
50 }

```

(ب)

ادامه شکل ۶۲- برنامه مدار رقص نور به کمک آرایه

در بخش setup در خطوط ۱۰ تا ۱۲ با یک حلقه for پایه‌های متصل به LEDهای چراغ راهنمای خودروها را به عنوان خروجی و در خطوط ۱۴ تا ۱۶ دو پایه مربوط به چراغ عابرین به عنوان خروجی معرفی شده‌اند. همچنین در خط ۱۸ پایه متصل به کلید به عنوان ورودی معرفی شده است.

در بخش loop ابتدا در خط ۲۴ یک شرط قرار دادیم تا هنگامی که کلید فشار داده شد وارد بلوک مربوطه شود و دستورات به ترتیب اجرا شود. دستورات این بلوک به ترتیب به این صورت اجرا می‌شود که ابتدا در چراغ راهنمای مربوط به خودروها چراغ سبز خاموش شده و چراغ زرد روشن می‌شود و بعد از حدود ۲ ثانیه چراغ زرد خاموش می‌شود (خط ۳۰). در ادامه با یک حلقه for که ۴ مرتبه تکرار می‌شود، چراغ قرمز روشن و خاموش می‌شود (خطوط ۳۲ تا ۳۷). در ادامه چراغ قرمز خودروها روشن و چراغ قرمز عابرین خاموش و چراغ سبز

عابرین روشن می‌شود (خطوط ۳۹ تا ۴۲). بعد از ۵ ثانیه چراغ‌ها به حالت اولیه بر می‌گردند.

۳ بُرد ESP32 را با کابل USB مناسب به کامپیوتر متصل کنید.

۴ در منوی TOOLS پورت اختصاص داده شده به بُرد ESP32 را مشخص کنید.

۵ دکمه آپلود را بزنید (یا کلیدهای ترکیبی Ctrl+U) و عملکرد مدار را بررسی کنید.

نکته

در این پروژه، در صورت نیاز به استفاده از لامپ‌ها یا چراغ‌های راهنمایی و رانندگی واقعی به جای LED، می‌توان با به‌کارگیری رله در مدار، خروجی‌های بُرد ESP32 را به صورت غیرمستقیم به بارهای با ولتاژ و جریان بالا متصل کرد.

پرسش

۱ اگر پایه سوئیچ را به جای اتصال به پایه ۳۷/۳ به زمین متصل کنیم چه تغییراتی باید در برنامه اعمال کرد؟
 ۲ اگر بخواهیم به جای اینکه فرمان از طریق کلید به چراغ راهنما داده شود به صورت خودکار عابر پیاده را تشخیص دهد و با حضور عابر در ابتدای خط عابر پیاده به طور خودکار سیستم فعال شود چه مداری پیشنهاد می‌دهید؟

توجه

جهت دسترسی به مدار شبیه‌سازی شده و کد برنامه به رمزینه سریع پاسخ اول پودمان چهارم رجوع کنید.



الزامات و نکات مهم برای جلوگیری از خطا در کدنویسی آردوینو

در برنامه‌نویسی زبان C در محیط Arduino IDE، رعایت نکات زیر برای جلوگیری از خطاها و نوشتن کدی تمیز و کارآمد ضروری است:

۱ حساسیت به حروف بزرگ و کوچک (Case Sensitivity)

زبان C به حروف بزرگ و کوچک حساس است. یعنی `LED`، `led` و `Led` سه متغیر کاملاً متفاوت هستند. نام متغیرها و توابع باید دقیقاً همان‌طور که تعریف شده‌اند، نوشته شوند.

۲ پایان‌بندی دستورات با نقطه‌ویرگول (;)

هر دستور در زبان C باید با علامت `;` تمام شود. فراموش کردن این علامت یکی از رایج‌ترین خطاهای نگارشی (Syntax Error) است که باعث می‌شود کامپایلر نتواند کد را بخواند.

۳ تفاوت علامت تساوی (=) و تساوی دوتایی (==)

■ از `=` برای "مقداردهی" (مثلاً `x = 5;`) استفاده می‌شود.

■ از `==` برای "مقایسه" در شرط‌ها (مثلاً `if(x == 5)`) استفاده می‌شود.

استفاده اشتباه از این دو می‌تواند باعث خطای منطقی شود (مثلاً شرط همیشه برقرار شود یا مقدار متغیر تغییر کند).

۴ استفاده صحیح از آرایه‌ها

■ آرایه‌ها فقط برای نگهداری مقادیر "هم‌نوع" (مثلاً فقط اعداد صحیح) هستند.

■ شماره‌گذاری آرایه‌ها از "صفر" شروع می‌شود.

اگر آرایه‌ای با اندازه ۴ داشته باشید، اندیس‌های معتبر ۰، ۱، ۲ و ۳ هستند. دسترسی به اندیس ۴ باعث خطا می‌شود.

۵ تعریف متغیرها قبل از استفاده

همه متغیرها باید قبل از استفاده در کد تعریف شوند (مثلاً `int x;`). همچنین بهتر است متغیرها در ابتدای بلوک کد (مثلاً داخل `setup` یا `loop`) تعریف شوند تا خوانایی کد حفظ شود.

۶ دقت در حلقه‌ها و شرط‌ها

■ در حلقه `for`، سه بخش (شروع، شرط، گام) باید با نقطه‌ویرگول از هم جدا شوند.

■ در حلقه `while` و `do-while`، شرط باید در پرانتز `()` قرار گیرد.

مراقب باشید حلقه‌ها به صورت بی‌نهایت (Infinite Loop) اجرا نشوند، مگر اینکه قصد توقف برنامه را داشته باشید.

۷ استفاده از کامنت‌ها

برای توضیح بخش‌های پیچیده کد، از کامنت‌ها استفاده کنید تا در آینده یا برای دیگران، هدف کد مشخص باشد.

■ کامنت تک خطی: `// توضیح`

■ کامنت چندخطی: `/* توضیح */`

رعایت این نکات باعث می‌شود کدهای شما بدون خطا کامپایل شده و عملکرد صحیحی در آردوینو داشته باشند.

خودارزیابی

- ۱ برای تبدیل دستورات انسان به زبانی قابل فهم توسط میکروکنترلر، به کدام ابزارها نیاز است؟
الف) فقط برد توسعه (ب) فقط کامپایلر
پ) برد توسعه و کامپایلر (ت) فقط زبان برنامه‌نویسی ساختاریافته
- ۲ به برنامه‌هایی که در محیط Arduino IDE نوشته می‌شوند، چه نامی داده می‌شود؟
الف) اسکچ (Sketch) (ب) پروگرامر (پ) درگاه USB (ت) کامپایلر
- ۳ برای ارسال برنامه به برد میکروکنترلری از طریق Arduino IDE، حتماً نیاز به استفاده از یک پروگرامر خارجی و خاص است.
☐ درست ☐ نادرست
- ۴ کدام دستور برای تنظیم یک پایه به عنوان ورودی یا خروجی استفاده می‌شود؟
الف) digitalWrite (Pin, Value) (ب) delay(time)
پ) pinMode(Pin, Mode) (ت) setup(Pin, Mode)
- ۵ دستور delay(time) مقدار دیجیتالی HIGH یا LOW را به یک پایه خروجی اعمال می‌کند.
☐ درست ☐ نادرست
- ۶ هدف اصلی استفاده از ساختارهای تصمیم‌گیری (شرطی) در برنامه‌نویسی میکروکنترلرها چیست؟
- ۷ کاربرد اصلی ساختار شرطی if ... else چیست؟
الف) اجرای دستورات به صورت خطی و پشت سرهم بدون توجه به شرط
ب) توقف برنامه به مدت زمان مشخص
پ) انجام یک عمل در صورت برقرار بودن شرط و عمل دیگری در صورت برقرار نبودن آن
د) تنظیم پایه‌ها به عنوان ورودی یا خروجی
- ۸ اگر شرط در ساختار if ... else برقرار نباشد، کدام بخش از کد اجرا می‌شود؟
- ۹ کدام یک از موارد زیر، دلیل اصلی استفاده از ساختارهای تکرار (حلقه) در برنامه‌نویسی میکروکنترلر است؟
الف) برای اینکه بتوانیم دستورات را به صورت خطی و بدون شرط اجرا کنیم
ب) برای جلوگیری از طولانی شدن کد و تسهیل تغییرات در تکرار یک عملیات چندباره
پ) برای اینکه فقط یک بار یک دستور اجرا شود
ت) برای تعیین مقدار اولیه پین‌های ورودی و خروجی
- ۱۰ در ساختار حلقه `for`، سه مؤلفه کلیدی که کنترل تکرار را بر عهده دارند، نام ببرید.
- ۱۱ در زبان C (آردوینو)، ساختار `while` زمانی استفاده می‌شود که تعداد تکرارها از قبل مشخص باشد (مثلاً شمارش از ۰ تا ۹)
☐ درست ☐ نادرست

۱۲ کدام یک از ساختارهای تکرار زیر، زمانی استفاده می‌شود که «حداقل یک بار» اجرای دستور ضروری باشد (مانند نمایش منوی اولیه)؟

الف) for ب) while پ) do-while ت) if

۱۳ چرا در مثال رقص نور، استفاده از آرایه برای ذخیره شماره پایه‌های خروجی (۱۳، ۱۲، ۱۴ و ۲۷) ضروری است؟

الف) زیرا آرایه‌ها سرعت پردازش میکروکنترلر را افزایش می‌دهند.

ب) زیرا شماره پایه‌ها به ترتیب عددی پشت‌سرهم نیستند و نمی‌توان با یک شمارنده ساده روی آنها حرکت کرد.

پ) زیرا آرایه‌ها اجازه می‌دهند تا از متغیرهای با نوع داده متفاوت استفاده کنیم.

ت) زیرا آرایه‌ها تنها برای ذخیره‌سازی مقادیر عددی کوچک (زیر ۱۰) کاربرد دارند.

۱۴ آرایه ساختاری است که برای نگهداری چند مقدار با نوع داده متفاوت (مثلاً همزمان عدد و متن) در یک نام واحد استفاده می‌شود.

□ درست □ نادرست

۱۵ در کد: `int LEDPins[۴] = {۱۳, ۱۲, ۱۴, ۲۷};` عدد داخل کروشه [۴] چه معنایی دارد؟

۱۶ کدام یک از موارد زیر، یکی از رایج‌ترین خطاهای نگارشی (Syntax Error) در زبان C است که باعث می‌شود کامپایلر نتواند کد را اجرا کند؟

الف) استفاده از متغیرهای غیرضروری ب) فراموش کردن نقطه‌ویرگول (؛) در انتهای دستورات

پ) استفاده از حلقه `for` به جای `while` ت) نام‌گذاری متغیرها با حروف بزرگ

۱۷ در زبان C، تفاوت بین علامت تساوی (=) برای مقداردهی و علامت تساوی دوتایی (==) برای مقایسه، بسیار مهم است و اشتباه گرفتن آنها می‌تواند منجر به خطای منطقی یا عدم اجرای صحیح شرط شود.

□ درست □ نادرست

۱۸ چرا در زبان C باید دقت کرد که نام متغیرها و توابع از نظر «حروف بزرگ و کوچک» (Case Sensitivity) دقیقاً همان‌طور که تعریف شده‌اند، نوشته شوند؟

ارزشیابی شایستگی برنامه‌نویسی میکروکنترلر

نام و نام خانوادگی	شماره ملی	تاریخ ارزشیابی:	نوبت:
کد حرفه	۸۲۱۲۱۲	حرفه:	موناژکاران تجهیزات و وسایل برقی و الکترونیکی
کد وظیفه شغل	۸۲۱۲۱۲۰۳	وظیفه شغل:	ساخت مدارهای دیجیتال و میکروکنترلر
کد کار	۸۲۱۲۱۲۰۳۴۲	کار:	انتخاب میکروکنترلر
کد برنامه‌نویسی	۸۲۱۲۱۲۰۳۴	کد پیمان/پودمان	کد پیمان/پودمان
کد سطح صلاحیت	۲	سطح صلاحیت	سطح صلاحیت
کد مهارت	مهارت	سطح شایستگی	سطح شایستگی

ردیف	مرا حل کار	شرایط انجام کار (ابزار، مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره‌دهی)	نتایج ممکن	نمره کسب شده
۱	نصب و راه‌اندازی محیط توسعه و سخت‌افزار توانایی نصب و پیکربندی نرم‌افزار Arduino IDE شناسایی و اتصال برد ESP32 به رایانه، نصب درایورها و انتخاب صحیح مدل برد و پورت ارتباطی (COM) در تنظیمات محیط توسعه.	رایانه یا لپ‌تاپ، نرم‌افزار Arduino IDE، برد ESP32، کابل USB، درایورهای لازم، دسترسی به اینترنت برای دانلود، میز کار، کارگاه الکترونیک یا سایت کامپیوتر، زمان تقریبی ۳۰-۲۰ دقیقه.	مشاهده عملکرد عملی فراگیر، چک‌لیست مراحل نصب و راه‌اندازی، بررسی تنظیمات IDE و اتصال موفق برد در نرم‌افزار	نصب IDE، شناسایی برد ESP32، نصب درایور و انتخاب صحیح برد و پورت COM به‌طور کامل و بدون خطا انجام مراحل با راهنمایی جزئی یا وجود خطای کوچک	۳ ۲ ۱	
۲	برنامه‌نویسی پایه در محیط Arduino توانایی نوشتن برنامه‌های ساده در محیط IDE با استفاده از دستورات پایه زبان C/C++ (مانند برنامه چشم‌کزن LED) و به‌کارگیری صحیح ساختار و دستورات زبان در پروژه‌های ساده.	رایانه یا لپ‌تاپ دارای Arduino IDE، برد ESP32، کابل USB، بردبرد، LED و مقاومت، منبع تغذیه مناسب، میز کار یا آزمایشگاه الکترونیک/کامپیوتر، زمان تقریبی ۳۰-۲۰ دقیقه.	مشاهده عملکرد عملی فراگیر، بررسی کد، نوشته‌شده در IDE، چک‌لیست ساختار برنامه (loop و setup) و دستورات پایه زبان C/C++	نوشتن صحیح برنامه ساده (مانند چشم‌کزن LED) با ساختار درست و استفاده مناسب از دستورات زبان نوشتن برنامه با خطاهای جزئی یا نیاز به راهنمایی محدود	۳ ۲ ۱	
۳	کامپایل، بارگذاری و رفع خطا (Compile/Check) توانایی بررسی (Compile/Check) کد، تشخیص و رفع خطاهای احتمالی و بارگذاری موفق برنامه بر روی برد ESP32	رایانه یا لپ‌تاپ دارای Arduino IDE، برد ESP32، کابل USB، برنامه نوشته‌شده در محیط IDE، دسترسی به رایانه در سایت یا کارگاه، زمان تقریبی ۱۵-۲۰ دقیقه.	مشاهده عملکرد عملی فراگیر، بررسی پیام‌های کامپایل در IDE، چک‌لیست مراحل رفع خطا، بارگذاری برنامه روی بردآماده یا بردبرد	کامپایل موفق کد، تشخیص و رفع خطاها و بارگذاری صحیح برنامه روی برد ESP32 بدون مشکل انجام مراحل با خطاهای جزئی یا نیاز به راهنمایی محدود	۳ ۲ ۱	
۴	کار با ورودی‌ها و خروجی‌ها توانایی اتصال و خواندن داده از ورودی‌ها (مانند دکمه یا سنسور) و کنترل خروجی‌ها (مانند LED) از طریق برنامه‌نویسی و پیاده‌سازی آن در پروژه‌های ساده کاربردی	رایانه یا لپ‌تاپ دارای Arduino IDE، برد ESP32، کابل USB، بردبرد، LED و مقاومت، دکمه فشاری یا سنسور ساده، سیم‌های اتصال، میز کار یا کارگاه الکترونیک، زمان تقریبی ۳۰-۲۰ دقیقه.	مشاهده عملکرد عملی فراگیر، بررسی نحوه سیم‌کشی مدار، بررسی کد برنامه در IDE، چک‌لیست خواندن صحیح ورودی و کنترل خروجی	اتصال صحیح مدار، خواندن درست ورودی (دکمه/سنسور) و کنترل دقیق خروجی (مانند روشن/خاموش شدن LED) از طریق برنامه انجام کار با خطاهای جزئی یا نیاز به راهنمایی محدود	۳ ۲ ۱	
شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش:						
		رعایت نسبی اصول ایمنی، بهداشت و ملاحظات زیست‌محیطی، با خطاهای جزئی در حین انجام کار.				
		عدم رعایت اصول ایمنی، بهداشت و ملاحظات زیست‌محیطی، یا بروز خطاهای جدی در نگرش و رفتار کاری.				

بلی ☐

خیر ☐

ارزشیابی کار (شایستگی انجام کار)

توضیحات:

۱- معیار شایستگی انجام کار:

- برای احراز شایستگی، کسب حداقل نمره ۲ از هر یک از ۴ شاخص (مراحل بحرانی) الزامی است.

- کسب حداقل نمره ۲ از بخش شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

- کسب حداقل میانگین ۲ از مراحل کار

۲- تعریف سطوح شایستگی: صرف نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

- مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

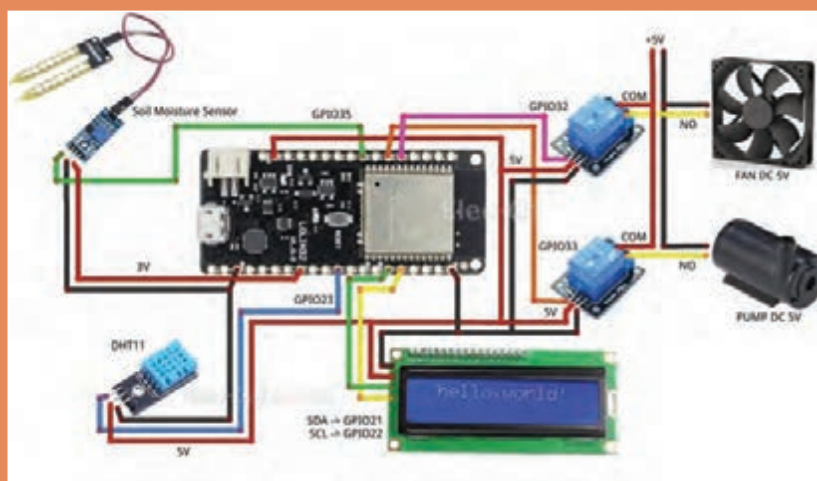
- تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.





پودمان پنجم

پیاده سازی مدارهای کاربردی با میکروکنترلر



در این پودمان، تعامل با محیط پیرامون و بهره‌گیری از حداکثر توانمندی‌های برد ESP32 را بررسی می‌کنیم. در واحد یادگیری اول، با یادگیری اصول ورودی‌ها و خروجی‌های دیجیتال، می‌آموزید چگونه با وضعیت پایه‌های GPIO، حسگرها و کلیدها را خوانده و تجهیزاتی مانند نمایشگرها و LEDها را کنترل کنید؛ این مهارت، مقدمه‌ای برای ورود به دنیای سیستم‌های کنترلی است. در واحد دوم، مفاهیم سیگنال‌های آنالوگ، مبدل ADC را تحلیل خواهید کرد. در این بخش یاد می‌گیرید چگونه با خواندن تغییرات محیطی توسط سنسورها و تنظیم جهت حرکت موتورها یا شدت نور، پروژه‌هایی هوشمند طراحی کنید. با تسلط بر این مباحث، با دیدی تحلیلی، مهارت لازم برای پیاده‌سازی سیستم‌های کاربردی و واقعی را کسب خواهید کرد. این پودمان، شما را از یک برنامه‌نویس مقدماتی به طراح سیستم‌های کنترلی توانمند تبدیل می‌کند. با تأکید بر دقت در سیم‌بندی و برنامه‌نویسی، بیاموزید چگونه ایده‌های خود را به واقعیت تبدیل کرده و گامی برای هوشمندسازی دنیای اطرافتان بردارید.

واحد یادگیری ۱: ورودی‌ها و خروجی‌های دیجیتال در میکروکنترلر ESP32

آیا تا به حال فکر کرده‌اید

- ۱ میکروکنترلر چگونه دنیای بیرون را حس می‌کند و به آن واکنش نشان می‌دهد؟
 - ۲ یک دستگاه هوشمند برای تصمیم‌گیری (مثلاً روشن کردن چراغ با تاریک شدن هوا)، چه اطلاعاتی را باید از محیط دریافت کند؟
 - ۳ چگونه می‌توان با استفاده از پایه‌های محدود یک میکروکنترلر، تجهیزات متنوعی مانند نمایشگرها و حسگرها را به‌طور هم‌زمان کنترل کرد؟
 - ۴ در طراحی پروژه‌های صنعتی یا هوشمندسازی خانه، تفاوت عملکردی ورودی و خروجی چیست و این دو چگونه در کنار هم یک سیستم را تشکیل می‌دهند؟
 - ۵ برای تبدیل یک ایده ساده به یک دستگاه کاربردی و عملیاتی، چه مراحل از برنامه‌نویسی و سیم‌بندی دقیق مورد نیاز است؟
- با مفاهیم پایه برنامه‌نویسی میکروکنترلر و محیط Arduino IDE آشنا هستید و یاد گرفته‌اید چگونه یک برنامه ساده را روی برد ESP32 اجرا کنید. همچنین توانسته‌اید با نوشتن چند برنامه ابتدایی، خروجی‌هایی مانند LED را کنترل کرده و وضعیت ورودی‌هایی نظیر کلید فشاری را بررسی کنید.
- در دنیای واقعی، میکروکنترلرها تنها به روشن و خاموش کردن یک LED محدود نمی‌شوند؛ بلکه وظیفه دارند اطلاعات محیط را دریافت کرده، پردازش کنند و سپس خروجی مناسب را فعال نمایند. این تعامل میان میکروکنترلر و دنیای پیرامون، از طریق ورودی‌ها و خروجی‌ها صورت می‌گیرد.
- در این واحد یادگیری، با طیف گسترده‌تری از ورودی‌ها و خروجی‌های دیجیتال آشنا خواهید شد و می‌آموزید که چگونه میان میکروکنترلر ESP32 و تجهیزاتی نظیر نمایشگرها، کلیدها، حسگرها و عملگرها ارتباط برقرار کنید.
- علاوه بر این، خواهید دید که چگونه می‌توان با ترکیب چندین ورودی و خروجی، یک سیستم کنترلی کارآمد طراحی کرد. هدف این واحد یادگیری، عبور از برنامه‌های مقدماتی و حرکت به سمت پروژه‌های کاربردی‌تر در صنعت و زندگی روزمره است. اجرای این پروژه‌ها اثر مستقیم بر ارزشیابی نهایی مهارتی خواهد داشت.

استاندارد عملکرد

مفاهیم ورودی/خروجی و GPIO در ESP32 را توضیح داده، پایه‌های دیجیتال را پیکربندی کند، با آنها حسگرها و نمایشگرها را کنترل کند و با نگرش سیستمی و دقت بالا، پروژه‌های کنترلی هوشمند را برنامه‌نویسی (C) و پیاده‌سازی نماید.

اتصال نمایشگرها به میکروکنترلر

در بسیاری از سیستم‌های الکترونیکی و کنترلی، تنها روشن یا خاموش شدن یک LED برای نمایش وضعیت سیستم کافی نیست، بلکه کاربر نیاز دارد اطلاعات بیشتری مانند عدد، متن، وضعیت یا پیام خطا را مشاهده کند.

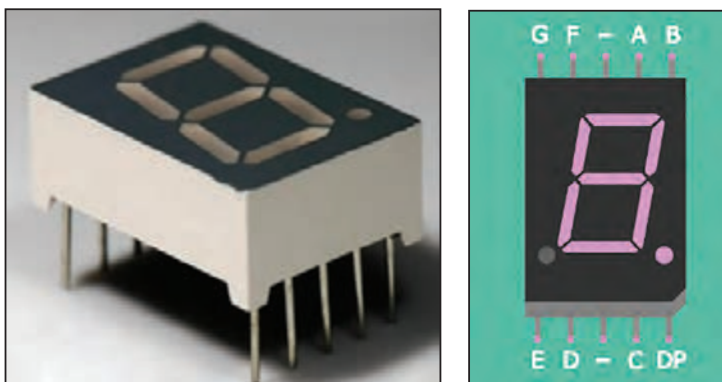
این وظیفه بر عهده نمایشگرها (Display) است. نمایشگرها، قطعات الکترونیکی خروجی هستند که می‌توانند اطلاعات پردازش شده توسط میکروکنترلر را به شکلی قابل فهم (عدد، حرف یا نماد) نمایش دهند. نمایشگرها بسته به نوع کاربرد، ساختار و میزان اطلاعاتی که نمایش می‌دهند، انواع مختلفی دارند. برخی نمایشگرها تنها قادر به نمایش عدد هستند و برخی دیگر می‌توانند متن و پیام‌های کامل را نمایش دهند. از نمایشگرها در دستگاه‌های مختلفی مانند ماشین حساب، ساعت دیجیتال، دستگاه‌های اندازه‌گیری، سیستم‌های صنعتی، لوازم خانگی و تجهیزات پزشکی استفاده می‌شود. در ادامه، با انواع پرکاربرد نمایشگرها آشنا می‌شوید و یاد می‌گیرید چگونه با استفاده از میکروکنترلر ESP32، اطلاعات مورد نظر را روی نمایشگرها نشان دهید.

انواع نمایشگرها

- سون سگمنت (Seven Segment): برای نمایش اعداد (۰ تا ۹) و بعضی از حروف و علائم
- نمایشگر LCD کاراکتری (مانند ۱۶×۲): برای نمایش متن، عدد و پیام
- نمایشگرهای گرافیکی (مانند OLED): برای نمایش متن، عدد و تصویر

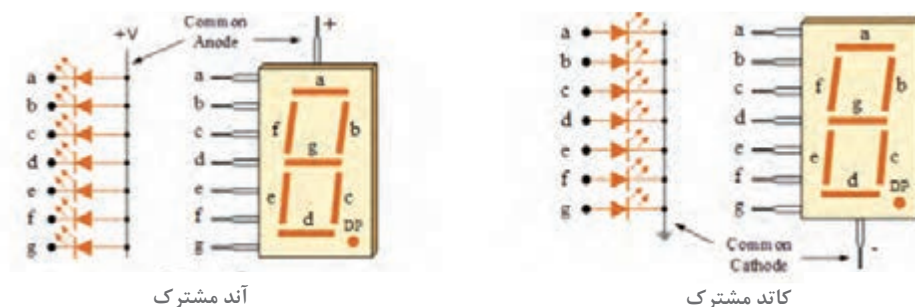
■ راه‌اندازی سون سگمنت با میکروکنترلر ESP32 مبتنی بر کتابخانه

■ **نمایشگر 7Segment:** نمایشگر سون سگمنت، از هفت LED تشکیل شده است که به شکل مستطیل مطابق شکل ۱ چیده شده‌اند. هر یک از این هفت LED یک قسمت (Segment) نامیده می‌شود و وقتی که یک قسمت روشن می‌شود، بخشی از یک رقم عددی نمایش داده می‌شود.



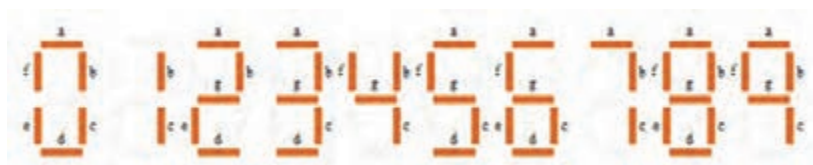
شکل ۱- نمایشگر 7Segment

یکی از پایه‌های هر کدام از هفت LED نمایشگر از بسته پلاستیکی مستطیلی بیرون آورده شده است. این پایه‌ها از a تا g نام‌گذاری می‌شوند و هر حرف نشان‌دهنده یک LED جداگانه است. پایه دوم LEDها به هم متصل شده و پایه مشترک را تشکیل می‌دهند. پایه مشترک نمایشگر نشان‌دهنده نوع نمایشگر سون سگمنت است و با توجه به اینکه پایه مشترک، آند LEDها یا کاتد آنها باشد، دو نوع نمایشگر سون سگمنت آند مشترک و کاتد مشترک ساخته می‌شود (شکل ۲).



شکل ۲- مدار داخلی نمایشگر 7Segment آند مشترک و کاتد مشترک

برای نمایش ارقام مختلف، با توجه به رقمی که باید نمایش داده شود، تعداد مشخصی از LEDها بایاس مستقیم می‌شوند. بنابراین، ارقام مختلف از صفر تا ۹ را می‌توان با استفاده از نمایشگر سون سگمنت نمایش داد. (شکل ۳)



شکل ۳- نمایش اعداد صفر تا ۹ با نمایشگر 7Segment مشترک

نمایش اعداد ۰ تا ۹ (شمارنده صعودی) با 7Segment

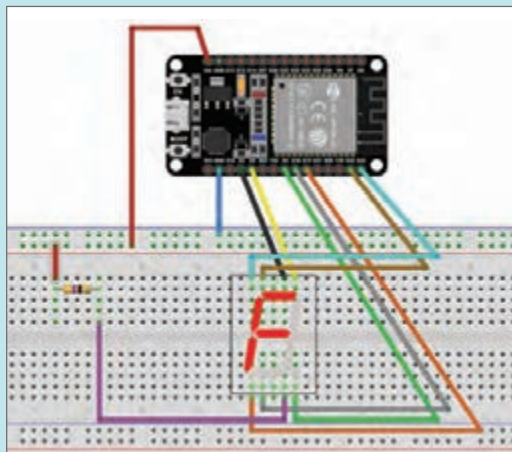
هدف پروژه: کسب مهارت در برنامه‌ریزی، شبیه‌سازی و برنامه‌نویسی سون سگمنت به عنوان یک نمایشگر عددی با استفاده از برنامه‌نویسی و به کارگیری کتابخانه مناسب آشنایی با دستور for. **مواد، ابزار و تجهیزات:** لوازم التحریر، دستورهای زبان C، برگه اطلاعات برد توسعه ESP32، برد توسعه ESP32، رایانه دارای برنامه Arduino IDE، شبیه‌ساز Wokwi، سون سگمنت کاتد مشترک یا آند مشترک (۱ عدد)، بردبرد، سیم جامپر، مقاومت ۴۷ اهم (۱ عدد).

مراحل اجرای کار:

۱ در شکل ۴ مدار شمارنده صعودی با 7Segment آند مشترک نمایش داده شده است. در این مدار زمان نمایش هر عدد توسط میکروکنترلر تعیین می‌شود. برای این منظور پایه‌های ۲۳، ۲۲، ۱۹، ۱۸، ۵، ۴ و ۲ میکروکنترلر برای اتصال به 7Segment در نظر گرفته شده است. مقاومت ۴۷ اهم برای کنترل جریان 7Segment مورد استفاده قرار می‌گیرد.

کار عملی ۱





شکل ۴- مدار شمارنده صعودی با 7Segment

۲ نصب کتابخانه: کنترل مستقیم بعضی از ورودی‌ها یا خروجی‌ها مانند نمایشگرها، سنسورها و... تنها با دستورهای ساده زبان C، کاری زمان‌بر و پیچیده است. برای ساده‌تر شدن برنامه‌نویسی، از مجموعه‌ای از کدهای آماده به نام کتابخانه (Library) استفاده می‌شود. کتابخانه‌ها شامل توابع و دستورهای هستند که کارهای پیچیده را به شکل ساده و قابل استفاده در اختیار برنامه‌نویس قرار می‌دهند. کتابخانه بسیاری از جزئیات سخت‌افزاری را مدیریت می‌کند و باعث ساده‌تر شدن برنامه‌نویسی می‌شود. به عنوان مثال با استفاده از کتابخانه نرم‌افزار می‌توان موارد زیر را به راحتی انجام داد:

■ کنترل پایه‌های متعدد یک سون سگمنت

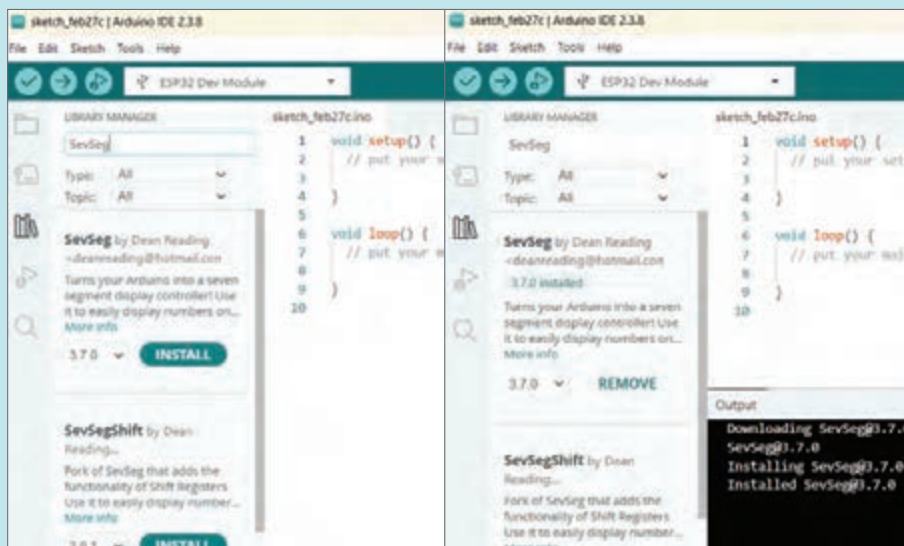
■ ارسال فرمان‌ها و داده‌ها به LCD

■ مدیریت زمان‌بندی و آدرس‌دهی

■ خواندن مقادیر سنسورها و پردازش داده‌های آنها

در برنامه‌نویسی مبتنی بر Arduino، کتابخانه‌ها در ابتدای برنامه فراخوانی می‌شوند. این کار باعث می‌شود توابع آماده کتابخانه در ادامه برنامه قابل استفاده باشند. در پروژه‌های این پودمان، از کتابخانه‌های مورد نیاز استفاده خواهد شد تا برنامه‌ها ساده‌تر و خواناتر نوشته شوند.

در این پروژه، به جای کنترل مستقیم سون سگمنت‌ها، از یک کتابخانه آماده به نام SevSeg استفاده شده است. نرم‌افزار Arduino IDE را فعال و مطابق شکل ۵ مراحل نصب کتابخانه را اجرا نمایید. در منوی سمت چپ، قسمت library manager را انتخاب نمایید و در قسمت جست‌وجو، کتابخانه SevSeg را تایپ نمایید. پس از پیدا شدن کتابخانه، آن را نصب کنید.



شکل ۵- مراحل نصب کتابخانه

۳ نرم افزار Arduino IDE را فعال نمایید و با زبان برنامه نویسی C برنامه شمارنده را بنویسید.

<pre>#include "SevSeg.h" SevSeg display; int counter = 0; void setup() { byte hardwareConfig = COMMON_ANODE; byte numDigits = 1; byte digitPins[] = {}; byte segmentPins[] = {15,2,4,5,18,19,22}; bool resistorsOnSegments = true; display.begin(hardwareConfig, numDigits, digitPins, segmentPins, resistorsOnSegments); display.setBrightness(90);} void loop() { display.setNumber(counter); display.refreshDisplay(); delay(1000); counter++; if (counter > 9) counter = 0; }</pre>	→ اضافه کردن کتابخانه به برنامه → تعریف یک نمونه از کتابخانه با نام 7Segment مشخص تعریف مشخصات 7Segment مورد استفاده (نوع، تعداد رقم، پایه های کنترلی، پین های متصل به ESP32 و به کارگیری مقاومت) پارامترهای قابل تعریف برای نمایشگر نمونه دستورات تعریف شده در کتابخانه
--	---

۴ از مسیر زیر وارد محیط شبیه سازی wokwi شده

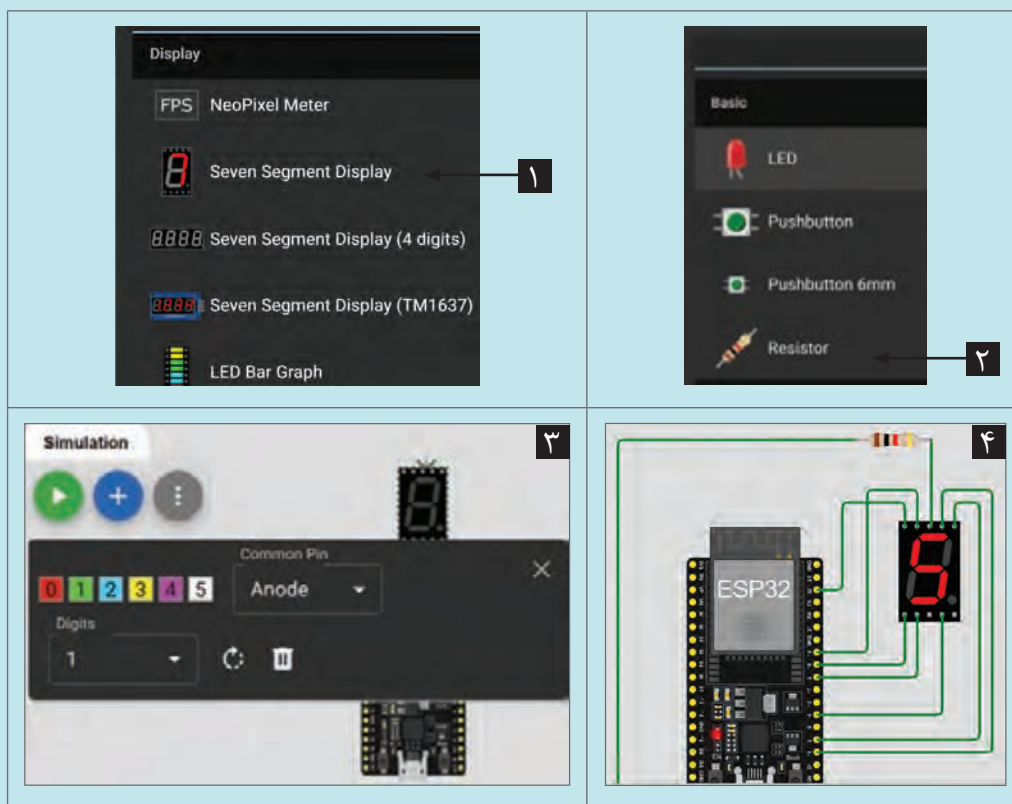
<https://wokwi.com/Project/new/ESP32>

علاوه بر مسیر فوق، می توان مطابق واحد یادگیری قبل نیز، وارد محیط شبیه سازی شد.

توجه

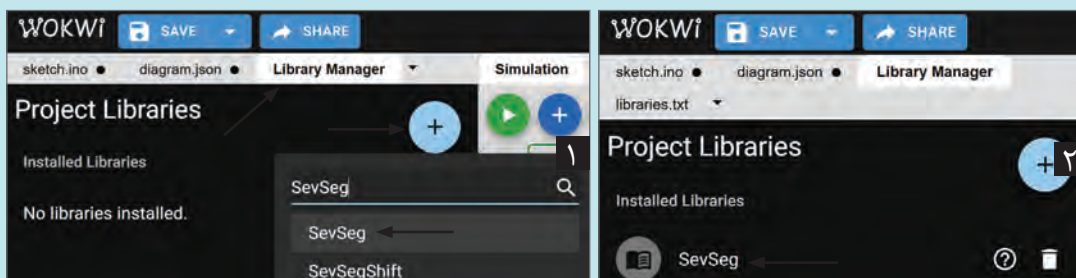


مطابق با شکل ۶ در قسمت شبیه ساز از کتابخانه 7Segment و مقاومت ۴۷ اهم را انتخاب نمایید و مدار را در شبیه ساز ببندید. با انتخاب 7Segment می توانید نوع و تعداد ارقام آن را مشخص نمایید.



شکل ۶

مطابق با شکل ۷، با انتخاب Library Manager و سپس +، نام کتابخانه مورد استفاده در برنامه را تایپ و جست‌وجو کنید تا در شبیه‌ساز، کتابخانه به برنامه اضافه شود.



شکل ۷

۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام نمایید. مدار را بر روی برد بسته و پروژه را اجرا نمایید.

۶ عملکرد مدار شمارنده را مشاهده کنید. در صورتی که مدار فعال نشد، اتصال پایه‌های میکروکنترلر و نوع 7Segment را کنترل کنید. در غیر این صورت به کمک مربی کارگاه، مدار را بازبینی و عیب‌یابی کنید. پس از رفع عیب، دوباره مدار را راه‌اندازی کنید.

۷ نتایج حاصل از راه اندازی مدار شمارنده را تحلیل کنید و به طور خلاصه بنویسید.

■ دستور **for**: برای اجرای حلقه‌های با تکرار مشخص، می‌توان از دستور **for** مطابق زیر استفاده کرد:

```
for (start stop;step){  
    مجموعه دستورات  
}
```

■ Stop در واقع شرط توقف حلقه می‌باشد. در برنامه بالا در قسمت **void loop()** عمل شمارش محدود

۰ تا ۹ به صورت تکرار شونده اجرا می‌شود. می‌توان به صورت ساده‌تر، کد زیر را جایگزین کرد.

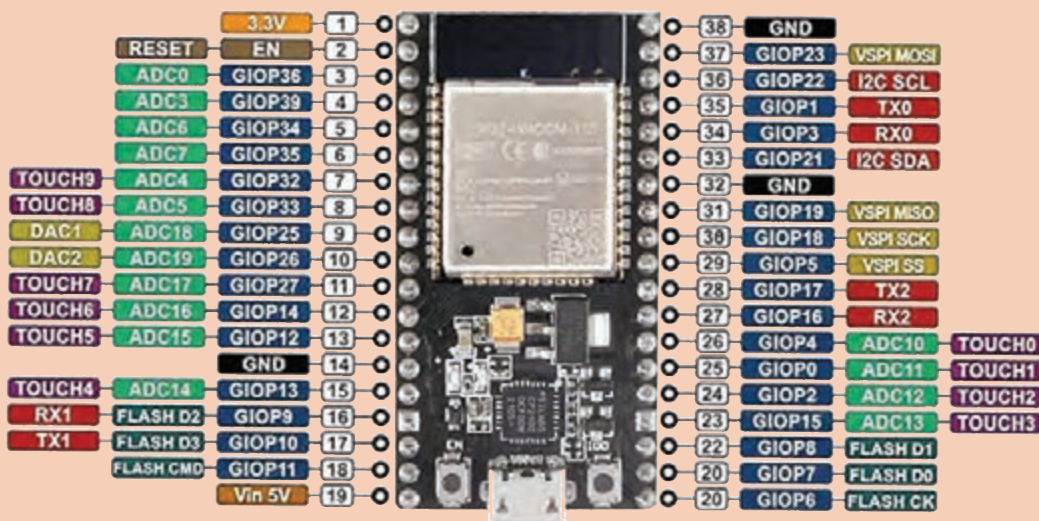
```
void loop() {  
    for (counter=0; counter<10; counter++)  
    {  
        display.setNumber(counter);  
        display.refreshDisplay();  
        delay(1000);  
    }  
}
```

تا زمانی که مقدار متغیر **counter** کوچک‌تر از ۱۰ باشد، حلقه تکرار می‌شود.

۸ برنامه را با حلقه **void loop()** بالا در شبیه‌ساز و اجرای عملی، تکرار کنید و نتیجه را بیان نمایید.

در صورت عدم دسترسی به مازول ESP32 ۳۰ پایه و استفاده از نسخه ۳۸ پایه در حین اجرای کارهای عملی هنجاریان باید دقت نظر خاص در تطبیق پایه‌ها و اتصالات با نقشه‌های عملی داشته باشند، زیرا تفاوت در چیدمان پایه (Pinout) می‌تواند به بروز خطا در عملکرد یا آسیب به قطعات شود. چیدمان پایه‌ها برای مازول ESP32 ۳۸ پایه در شکل ۸ نشان داده شده است.

نکته مهم



شکل ۸- مازول ESP32 ۳۸ پایه



کنترل نمایش اعداد ۰ تا ۹ در 7Segment با ۲ کلید به وسیله ESP32

هدف پروژه: کسب مهارت در برنامه‌ریزی، شبیه‌سازی و برنامه‌نویسی کنترل عدد روی سون سگمنت به عنوان یک نمایشگر عددی توسط دو کلید.

مواد، ابزار و تجهیزات: لوازم التحریر، دستورهای زبان C، برگه اطلاعات برد توسعه ESP32، برد توسعه ESP32، رایانه دارای برنامه Arduino IDE، شبیه‌ساز Wokwi، سون سگمنت (کاتد مشترک یا آند مشترک) یک عدد، برد بُرد، سیم جامپر، مقاومت ۴۷ اهم ۱ عدد و دو کلید فشاری.

مراحل اجرای کار:

۱ در شکل ۹ مدار کنترل عدد و نمایش آن بر روی 7Segment آند مشترک نمایش داده شده است. در این مدار یک کلید، عدد را ۱ واحد افزایش می‌دهد و کلید دیگر ۱ واحد کاهش می‌دهد. برای این منظور پین‌های (۱۵، ۲، ۴، ۵، ۱۸، ۱۹، ۲۲) از میکروکنترلر برای اتصال به 7Segment و پین‌های ۱۲ و ۱۳ برای کلیدها در نظر گرفته شده است. مقاومت ۴۷ اهم برای کنترل جریان 7Segment لحاظ شده است.

۲ نرم‌افزار Arduino IDE را فعال نمایید و با زبان برنامه‌نویسی C برنامه را بنویسید.

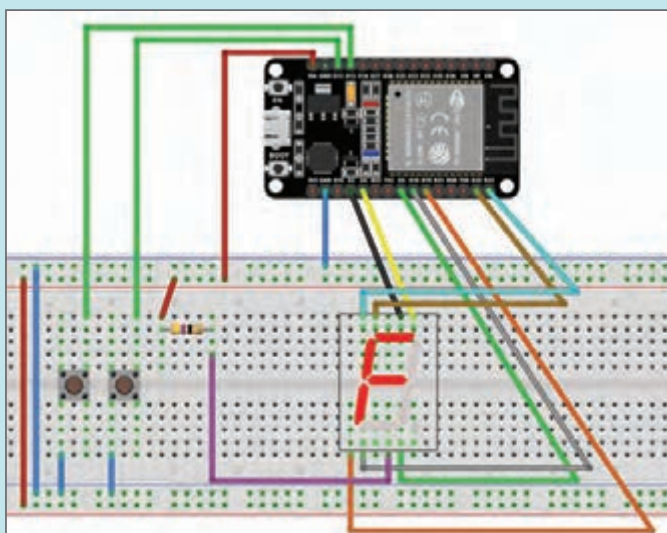
۳ با توجه به دستورات آموزش داده شده، برنامه را تحلیل نمایید و در کلاس گزارش دهید.

۴ از مسیر زیر وارد محیط شبیه‌سازی wokwi شده

<https://wokwi.com/Projects/new/ESP32>

توجه

علاوه بر مسیر فوق، می‌توان مطابق واحد یادگیری قبل نیز، وارد محیط شبیه‌سازی شد.



شکل ۹

۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام نمایید. مدار را بر روی برد بُرد بسته و پروژه را اجرا نمایید.


```
#include "SevSeg.h"
```

```
SevSeg display;
```

```
int Upbotton = 12;
```

```
int downbutton =13;
```

```
int counter = 0;
```

```
void setup() {
```

```
    byte hardwareConfig = COMMON_ANODE;
```

```
    byte numDigits = 1;
```

```
    byte digitPins[] = {};
```

```
        byte segmentPins[] = {15,2,4,5,18,19,22};
```

```
        bool resistorsOnSegments = true;
```

```
        display.begin(hardwareConfig, numDigits,  
digitPins, segmentPins, resistorsOnSegments);
```

```
        display.setBrightness(90);
```

```
        pinMode(Upbotton, INPUT_PULLUP);
```

```
        pinMode(downbutton, INPUT_PULLUP);
```

```
    }
```

```
void loop() {
```

```
    if (digitalRead(Upbotton) == LOW) {
```

```
        if (counter < 10)
```

```
            counter++;
```

```
    if (digitalRead(downbutton) == LOW) {
```

```
        if (counter > 0)
```

```
            counter--;
```

```
    display.setNumber(counter);
```

```
    display.refreshDisplay();
```

```
    delay(20);    }
```

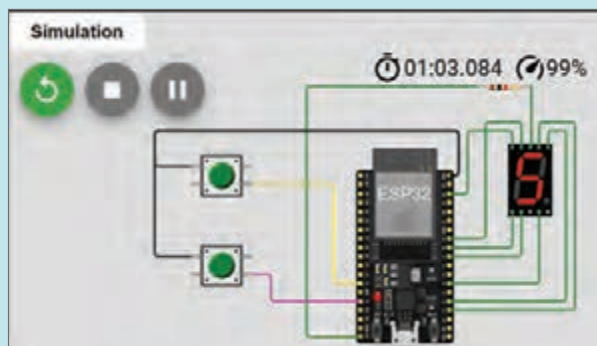
اضافه کردن کتابخانه به برنامه
تعریف یک نمونه از کتابخانه با نام مشخص

تعریف مشخصات 7Segment مورد استفاده (نوع، تعداد رقم، پایه‌های کنترلی، پین‌های متصل به ESP32 و به کارگیری مقاومت)

پارامترهای قابل تعریف برای نمایشگر

تعریف کلیدها

نمونه دستورات تعریف شده در کتابخانه



۶ عملکرد مدار را مشاهده کنید.
۷ نتایج حاصل از راه‌اندازی مدار را تحلیل کنید و به‌طور خلاصه بنویسید.

شکل ۱۰

تمرین‌های زیر را با راهنمایی هنرآموز و به‌صورت گروهی طراحی کرده و به‌صورت نرم‌افزاری (wokwi) و عملی اجرا نمایید.

۱ برنامه یک شمارنده نزولی (۹ تا صفر) به‌صورت تکرار شونده را بر روی 7Segment بنویسید. سپس به‌صورت شبیه‌سازی در Wokwi و عملی اجرا نمایید.

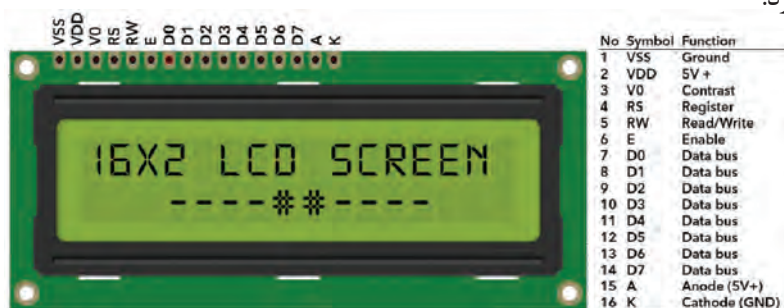
۲ برنامه یک شمارنده صعودی - نزولی (صفر تا ۹۹ و سپس ۹ تا صفر به‌صورت تکرار شونده) را بر روی 7Segment بنویسید. سپس به‌صورت شبیه‌سازی در Wokwi و عملی اجرا نمایید.

پروژه



نمایشگر LCD (Liquid Crystal Display)

نمایشگر LCD یکی از پرکاربردترین ابزارهای خروجی در پروژه‌های الکترونیکی و میکروکنترلی است که برای نمایش اعداد، حروف، کلمات و اطلاعات متنی استفاده می‌شود. این نمایشگرها به دلیل مصرف انرژی کم، قیمت مناسب، ضخامت کم و خوانایی خوب، به‌طور گسترده در دستگاه‌های صنعتی و آموزشی به کار می‌روند. LCDهای کاراکتری معمولاً در مدل‌های ۱۶×۲ (شانزده ستون و دو سطر) و ۲۰×۴ عرضه می‌شوند که مدل ۱۶×۲ یکی از رایج‌ترین گزینه‌ها در پروژه‌های آموزشی است (شکل ۱۱). این نوع LCD قابلیت نمایش متن، عدد، کاراکترهای خاص و حتی شکل‌های ساده را دارد. نمایشگرهای LCD استاندارد دارای ۱۶ پایه هستند که هر یک وظیفه مشخصی مانند تغذیه، کنترل و انتقال داده را بر عهده دارند. آشنایی با این پایه‌ها نقش مهمی در راه‌اندازی صحیح LCD دارد.



شکل ۱۱

LCD های کاراکتری قابلیت راه اندازی در دو حالت ۸ بیتی و ۴ بیتی را دارند. در حالت ۸ بیتی (D۰ تا D۷)، برای انتقال داده ها به تعداد بیشتری از پایه های میکروکنترلر نیاز است، در حالی که در حالت ۴ بیتی (D۴ تا D۷) تنها از چهار پایه داده استفاده می شود. با توجه به محدود بودن تعداد پایه های میکروکنترلر و ساده تر بودن اتصالات، در این کتاب تمام پروژه ها در مد ۴ بیتی اجرا می شوند. همچنین برای ساده سازی فرایند برنامه نویسی و کاهش خطا، از کتابخانه های آماده استفاده می شود.

■ کتابخانه LiquidCrystal: کتابخانه LiquidCrystal شامل مجموعه ای از توابع آماده است که امکان کنترل آسان LCD را فراهم می کند. مهم ترین توابع این کتابخانه در جدول ۱ معرفی شده اند.

جدول ۱

تابع	عملکرد
LiquidCrystal()	تعیین اتصالات میکروکنترلر به (LCD مد ۴ یا ۸ بیتی)
lcd.begin()	راه اندازی LCD با تعیین تعداد سطر و ستون
lcd.clear()	پاک کردن محتوای نمایشگر
lcd.home()	انتقال مکان نما به موقعیت (۰,۰)
lcd.setCursor(col, row)	تعیین محل نمایش اطلاعات
lcd.write()	نمایش یک کاراکتر
lcd.print()	نمایش یک رشته (متن)
lcd.cursor() / lcd.noCursor()	فعال / غیرفعال کردن مکان نما
lcd.blink() / lcd.noBlink()	فعال / غیرفعال کردن چشمک زدن مکان نما
lcd.display() / lcd.noDisplay()	روشن / خاموش کردن نمایش اطلاعات
lcd.scrollDisplayLeft() / lcd.scrollDisplayRight()	جابجایی متن به چپ یا راست
lcd.autoscroll() / lcd.noAutoscroll()	حرکت خودکار متن
lcd.leftToRight() / lcd.rightToLeft()	تنظیم جهت نوشتن متن
lcd.createChar()	تعریف کاراکتر دلخواه (حداکثر ۸ عدد)

راه اندازی LCD با میکروکنترلر ESP32 مبتنی بر کتابخانه

هدف پروژه: در این پروژه، هنرجو یاد می گیرد که LCD کاراکتری ۱۶×۲ را به ESP32 متصل کند و با استفاده از کتابخانه LiquidCrystal، متن دلخواه را در سطر اول و دوم LCD نمایش دهد.

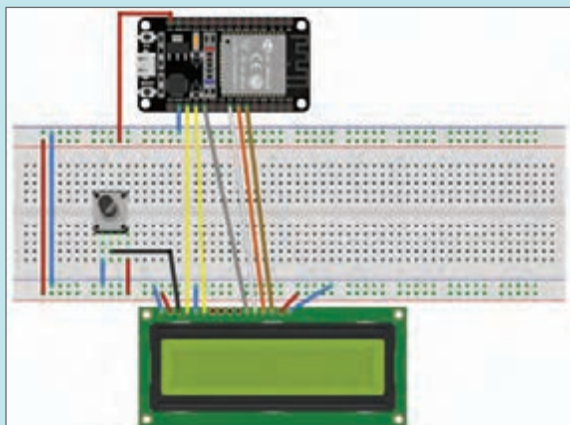
مواد، ابزار و تجهیزات: لوازم التحریر، دستورهای زبان C، برگه اطلاعات برد توسعه ESP32، برد توسعه ESP32، رایانه دارای برنامه Arduino IDE، شبیه ساز Wokwi، LCD کاراکتری ۱۶×۲، پتانسیومتر ۱۰kΩ، برد برد، سیم جامپر.

کار عملی ۳



مراحل اجرای کار:

۱ در این پروژه عبارت‌های *ESP32* و {ARDUINO} را بر روی سطر اول و دوم LCD نمایش



شکل ۱۲

می‌دهیم. مطابق با شکل ۱۲، سخت‌افزار مورد نیاز را آماده می‌کنیم. سپس مبتنی بر توابع و دستورات کتابخانه، برنامه را می‌نویسیم. در راه‌اندازی LCD از خروجی ولتاژ ۵ ولت (Vin) میکروکنترلر استفاده می‌کنیم. به جهت تنظیم وضوح کاراکترهای نمایش داده شده بر روی LCD، از یک پتانسیومتر $10K\Omega$ متصل به پایه V_0 استفاده می‌شود. پایه‌های (۱۴, ۲۷, ۲۶, ۲۵, ۳۲, ۳۳) برای اتصال LCD به ESP32 در نظر گرفته شده است.

۲ مطابق با توضیحات ارائه شده در کار عملی ۱، کتابخانه LiquidCrystal را هم در Arduino IDE و هم در شبیه‌ساز Wokwi نصب نمایید.

با توجه به این که در این کاربرد تنها عملیات نوشتن روی LCD انجام می‌گیرد، اتصال مستقیم پایه RW به زمین (GND) پیشنهاد می‌شود تا پایه کمتری از ESP32 به کار گرفته شود و منابع میکروکنترلر بهینه مصرف شوند.

نکته مهم



۳ نرم‌افزار Arduino IDE را فعال نمایید و با زبان برنامه‌نویسی C برنامه زیر را بنویسید. با توجه به توضیح دستورات، عملکرد هر خط از برنامه را در شکل زیر بنویسید.

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(14, 27, 26, 25, 33, 32);
void setup() {
  lcd.begin(16, 2);
  lcd.setCursor(0, 0);
  lcd.print("**ESP32*");
  lcd.setCursor(0, 1);
  lcd.print("ARDUINO");
}
void loop() {}
```

در کد

LiquidCrystal lcd (۱۴, ۲۷, ۲۶, ۲۵, ۳۳, ۳۲)

پایه‌ها به ترتیب زیر مشخص می‌شوند.

LiquidCrystal lcd (RS, EN, D۴, D۵, D۶, D۷);

۴ از مسیر زیر وارد محیط شبیه‌سازی wokwi شده

[/https://wokwi.com/Projects/new/ESP32](https://wokwi.com/Projects/new/ESP32)

۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام نمایید. مدار را بر روی بردبرد بسته و پروژه را اجرا نمایید.

۶ عملکرد مدار را مشاهده کنید.

۷ مقدار ولوم متصل به LCD را تغییر دهید و نتیجه را گزارش دهید.

۸ نتایج حاصل از راه‌اندازی مدار را تحلیل کنید و به‌طور خلاصه بنویسید.

تمرین‌های زیر را با راهنمایی هنرآموز و به‌صورت گروهی طراحی کرده و به‌صورت نرم‌افزاری (wokwi) و عملی اجرا نمایید.

■ عبارت‌های نمایش داده شده را تغییر دهید و نام و نام خانوادگی خود را نمایش دهید.

■ پروژه قبل را به‌صورت چشمک‌زن نمایش دهید.

■ کد را به‌گونه‌ای تغییر دهید که ابتدا یک پیام خوش‌آمدگویی در سطر دوم به مدت ۲ ثانیه ظاهر شود و سپس نام و نام خانوادگی شما در وسط سطر اول و دوم نمایش داده شود و هر ۲ ثانیه جای آنها عوض شود.

تکلیف



دریافت متن از Serial Monitor و نمایش روی LCD

هدف پروژه: کسب مهارت در برنامه‌نویسی، شبیه‌سازی، پروگرام کردن میکروکنترلر و مدار سخت‌افزاری شامل LCD و برقراری ارتباط بین رایانه و میکروکنترلر و نمایش داده‌ها بر روی LCD

مواد، ابزار و تجهیزات: لوازم‌التحریر، دستورهای زبان C، برگه اطلاعات برد توسعه ESP32، برد توسعه ESP32، رایانه دارای برنامه Arduino IDE، شبیه‌ساز Wokwi، LCD کاراکتری ۱۶×۲، پتانسیومتر ۱۰kΩ، بردبرد، سیم جامپر.

مراحل اجرای کار:

۱ در این پروژه، هنرجو ارتباط سریال بین ESP32 و رایانه را برقرار کرده و داده متنی ارسال شده از Serial Monitor را دریافت و بر روی LCD کاراکتری ۱۶×۲ نمایش می‌دهد. دقت شود که در صورتی که طول متن بیشتر از ظرفیت LCD باشد، فقط بخشی از آن نمایش داده می‌شود.

■ **ارتباط سریال:** ارتباط سریال یکی از روش‌های ارسال و دریافت اطلاعات بین میکروکنترلر و رایانه است. در محیط Arduino IDE، ابزار Serial Monitor امکان ارسال متن از رایانه به میکروکنترلر و دریافت پاسخ را فراهم می‌کند. توابع زیر در ارتباط سریال استفاده می‌شود:

□ تابع Serial.begin() برای راه‌اندازی ارتباط سریال استفاده می‌شود.

□ تابع Serial.available() بررسی می‌کند آیا داده‌ای دریافت شده است یا خیر.

□ تابع Serial.readString() متن ارسال شده را دریافت می‌کند.

مطابق با شکل ۱۲، سخت‌افزار مورد نیاز را آماده نمایید. سپس با استفاده از توابع و دستورات، برنامه را می‌نویسیم.

کار عملی ۴



۲ مطابق با توضیحات ارائه شده در کار عملی ۱، کتابخانه LiquidCrystal را هم در Arduino IDE و هم در شبیه‌ساز Wokwi نصب نمایید.

۳ نرم‌افزار Arduino IDE را فعال نمایید و با زبان برنامه‌نویسی C برنامه زیر را بنویسید. با توجه به توضیح دستورات، عملکرد هر خط از برنامه را در شکل زیر بنویسید.

#include <LiquidCrystal.h>	//	
LiquidCrystal lcd(14, 27, 26, 25, 33, 32);	//	
String receivedText = "";	//	
void setup() {		
lcd.begin(16, 2);	//	
lcd.clear();	//	
Serial.begin(9600);	//	نرخ انتقال داده (Baud Rate) بین رایانه و Esp32
lcd.setCursor(0, 0);	//	بر روی ۹۶۰۰ بیت بر ثانیه تنظیم می‌گردد.
lcd.print("Send text from");	//	
lcd.setCursor(0, 1);	//	
lcd.print("Serial Monitor");	//	
void loop() {		
if (Serial.available() > 0)	//	
{		
receivedText = Serial.readString();	//	
lcd.clear();		
lcd.setCursor(0, 0);		
lcd.print(receivedText);	//	
}		
}}		

۴ از مسیر زیر وارد محیط شبیه‌سازی wokwi شده

<https://wokwi.com/Projects/new/ESP32>

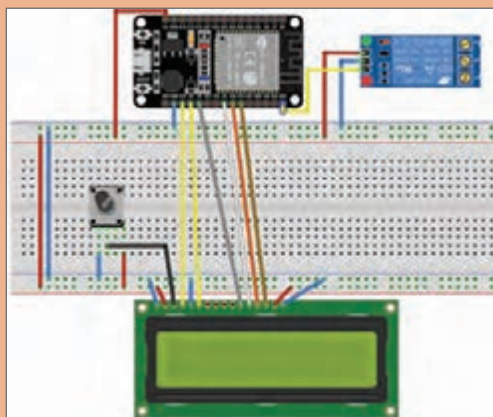
۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام نمایید. مدار را بر روی بردبرد بسته و پروژه را اجرا نمایید.

۶ عملکرد مدار را مشاهده کنید. در صورتی که مدار فعال نشد، اتصال پایه‌های میکروکنترلر را کنترل کنید. در غیر این صورت به کمک مربی کارگاه، مدار را بازبینی و عیب‌یابی کنید. پس از رفع عیب، دوباره مدار را راه اندازی کنید.

۷ نتایج حاصل از راه‌اندازی مدار را تحلیل کنید و به‌طور خلاصه بنویسید.



تمرین‌های زیر را با راهنمایی هنرآموز و به صورت گروهی طراحی کرده و به صورت نرم‌افزاری (wokwi) و عملی اجرا نمایید.



شکل ۱۳

- مطابق با شکل ۱۳ برنامه‌ای بنویسید که:
 - ابتدا پیام "Enter password..." را در سطر اول چاپ کند.
 - پس از دریافت رمز (شامل عدد، کاراکتر و حروف) از طریق Serial Monitor، در صورت صحیح بودن، پیام "password true" چاپ و یک رله (یا LED) فعال شود.
 - و در غیر این صورت پیام "password wrong!" را ۲ ثانیه نمایش دهد.

■ برنامه‌ای بنویسید که یک رشته از طریق Serial Monitor دریافت کند و به صورت متحرک از سمت چپ به راست نمایش دهد.



نمایش زمان و تاریخ بر روی LCD

هدف پروژه: کسب مهارت در برنامه‌نویسی، شبیه‌سازی، پروگرام کردن میکروکنترلر و مدار سخت‌افزاری شامل LCD و نمایش زمان و تاریخ بر روی LCD

مواد، ابزار و تجهیزات: لوازم التحریر، دستوره‌های زبان C، برد توسعه ESP32 و برگه اطلاعات برد، رایانه دارای برنامه Arduino IDE، شبیه‌ساز Wokwi، LCD کاراکتری ۱۶×۲، پتانسیومتر ۱۰kΩ، بردبرد، جامپر.

مراحل اجرای کار:

- ۱ در این پروژه، زمان شامل ساعت، دقیقه و ثانیه می‌باشد و هر ثانیه به‌روزرسانی می‌شود. همچنین تاریخ نیز به صورت روز، ماه و سال شبیه‌سازی شده و پس از گذشت ۲۴ ساعت، یک روز افزایش یابد (شکل ۱۴). از دستور delay (۱۰۰۰) برای شبیه‌سازی گذر یک ثانیه و از متغیرهای عددی برای نگهداری ساعت، دقیقه، ثانیه، روز، ماه و سال استفاده می‌شود. خروجی مورد انتظار روی LCD، به عنوان مثال به صورت شکل ۱۴ می‌باشد. مطابق با شکل ۱۲، سخت‌افزار مورد نیاز را آماده کرده سپس مبتنی با توابع و دستورات، برنامه را می‌نویسیم.



شکل ۱۴

- ۲ مطابق با توضیحات ارائه شده در کار عملی ۱، کتابخانه LiquidCrystal را هم در Arduino IDE و هم در شبیه‌ساز Wokwi نصب نمایید.

۳ نرم‌افزار Arduino IDE را فعال نمایید و با زبان برنامه‌نویسی C برنامه زیر را بنویسید. با توجه به توضیح دستورات، عملکرد هر خط از برنامه را در شکل زیر بنویسید.

<pre>#include <LiquidCrystal.h> LiquidCrystal lcd(14,27,26,25,33,32); int hour = 12; int minute = 0; int second = 0; int day = 1; int month = 1; int year = 1403; void setup(){ lcd.begin(2,16) lcd.clear();} void loop() { lcd.setCursor(0,0) lcd.print("Time: "); lcd.print(hour); lcd.print(":"); lcd.print(minute); lcd.print(":"); lcd.print(second); lcd.setCursor(1,0) lcd.print("Date: "); lcd.print(day); lcd.print("/"); lcd.print(month);</pre>	<pre>lcd.print("/"); lcd.print(year); delay(1000) ; second++; if (second == 60) { second = 0; minute++;} if (minute == 60) { minute = 0; hour++; } if (hour == 24) { hour = 0; day++;} if (month <= 6 && day == 32){ day = 1; month++; } if (month > 6 && day == 31) { day = 1; month++; } if (month == 13) { month = 1; year++;} lcd.clear();}</pre>
--	--

۴ در صفحه وب <https://docs.wokwi.com> وارد قسمت Guides/ESP۳۲ Simulator شده و در قسمت Arduino Core قالب خام شبیه‌ساز (Arduino_ESP۳۲ Project Template) را انتخاب نمایید. سپس مطابق با شکل ۱۲، سخت‌افزار را بسته و با اجرای کد، شبیه‌سازی نمایید.

۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام نمایید. مدار را بر روی برد برد بسته و پروژه را اجرا نمایید.

۶ عملکرد مدار را مشاهده کنید.

۷ نتایج حاصل از راه‌اندازی مدار را تحلیل کنید و به‌طور خلاصه بنویسید.

۸ در برنامه از عملگر منطقی && استفاده شده است. با راهنمایی هنرآموز خود، در مورد علت به‌کارگیری این عملگر توضیح دهید.

اتصال سنسورهای دیجیتال به میکروکنترلر

برای اینکه یک سیستم هوشمند بتواند تصمیم بگیرد، ابتدا باید اطلاعاتی از محیط اطراف خود دریافت کند. این اطلاعات می‌توانند شامل دما، رطوبت، نور، فاصله، حرکت و یا فشار باشند. قطعه‌ای که وظیفه دریافت این اطلاعات را بر عهده دارد، سنسور نامیده می‌شود. سنسورها نقش حواس یک سیستم هوشمند را ایفا می‌کنند و بدون آنها، میکروکنترلر قادر به درک شرایط محیط نخواهد بود. در جدول زیر مثال‌هایی از کاربرد سنسورها ارائه شده است:

نوع سنسور	کاربرد
سنسور مادون قرمز (IR)	تشخیص حرکت، رنگ، فاصله، شمارش، اندازه‌گیری سرعت
دما و رطوبت (DHT11 / DHT22)	کنترل محیط گلخانه، کولر، رطوبت سنج
فاصله سنج (HC-SR04)	جلوگیری از برخورد ربات‌ها، اندازه‌گیری ارتفاع، پارکینگ هوشمند
نور (LDR)	تنظیم روشنایی اتوماتیک، پنل خورشیدی
حرکت (PIR)	سیستم هشدار، چراغ خودکار
رطوبت سنج خاک (moisor)	گلخانه

■ نقش سنسورها در سیستم‌های میکروکنترلری

- دریافت داده‌های واقعی از محیط
- امکان تصمیم‌گیری هوشمند براساس شرایط محیطی
- اتصال به میکروکنترلر برای پردازش و نمایش اطلاعات
- فعال کردن خروجی‌ها مانند LED و موتور براساس شرایط مشخص

■ سنسور مادون قرمز (IR)

سنسور مادون قرمز ابزاری الکترونیکی است که برای حس کردن ویژگی‌های خاصی در محیط اطراف استفاده می‌شود. این کار با انتشار تابش مادون قرمز انجام می‌شود. حسگرهای مادون قرمز همچنین قادر به اندازه‌گیری گرمای ساطع شده توسط جسم و تشخیص حرکت نیز هستند. فناوری سنسور مادون قرمز نه تنها در صنعت بلکه در زندگی روزمره نیز یافت می‌شود. به عنوان مثال تلویزیون‌ها از یک سنسور مادون قرمز برای تفسیر سیگنال‌های ارسالی از کنترل از راه دور استفاده می‌کنند.



ماژول IR سه پایه

ماژول IR چهارپایه

ماژول IR (فقط گیرنده)

شکل ۱۵- انواع ماژول سنسور مادون قرمز

مطابق با شکل ۱۵، حسگرهای مادون قرمز دارای یک گیرنده و یک فرستنده هستند. یکی از LEDهای آن ساطع کننده نور مادون قرمز است (فرستنده) و زمانی که دستان را جلوی آن قرار می‌دهیم نور به دست ما برخورد می‌کند و برمی‌گردد و توسط LED دیگر (گیرنده) دریافت می‌شود. این سنسور همچنین می‌تواند فقط شامل گیرنده باشد. در این حالت نور مادون قرمز ساطع شده از محیط را دریافت می‌کند و خروجی آن فعال می‌شود که در کاربردهایی مانند تشخیص آتش و ... کاربرد دارد. یک ولوم روی آن قرار دارد که با چرخاندن آن می‌توانیم شدت حساسیت به فاصله را تنظیم کنیم. این ماژول در دو مدل سه پایه و چهار پایه ارائه می‌شود. مدل سه پایه شامل VCC، GND و DO (خروجی دیجیتال) می‌باشد و در مدل چهار پایه، خروجی آنالوگ (AO) هم وجود دارد.

از کاربردهای مهم دیگر این سنسور به موارد زیر می‌توان اشاره کرد:

- رباتیک: تشخیص خط، مانع و مسیر
- کنترل از راه دور (ریموت کنترل): حسگرهای گیرنده در تلویزیون، کولر و ...
- سامانه‌های امنیتی و حفاظتی: حسگر درب پارکینگ، آسانسور و ...
- اندازه‌گیری سرعت و شمارش: اندازه‌گیری سرعت چرخ موتور و ...
- پزشکی و تجهیزات زیستی: پالس اکسیمتر، تشخیص جریان خون و ...
- صنعت - نوار نقاله و اتوماسیون: تشخیص عبور محصول، شمارش قطعات و ...
- خودرو: حسگر حضور دست روی فرمان و ...

اتصال حسگر مادون قرمز (IR) به میکروکنترلر

هدف پروژه: کسب مهارت در برنامه‌نویسی، شبیه‌سازی، پروگرام کردن میکروکنترلر و مدار سخت‌افزاری شامل حسگر IR

مواد، ابزار و تجهیزات: لوازم التحریر، دستوره‌های زبان C، برد توسعه ESP32 و برگه اطلاعات برد، رایانه دارای برنامه Arduino IDE، شبیه‌ساز Wokwi، LED، بردبرد، جامپر و سنسور IR.

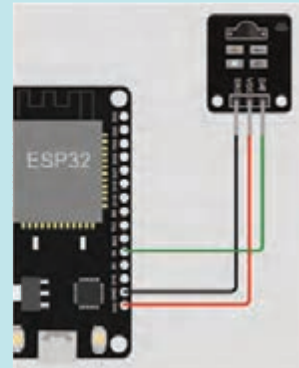
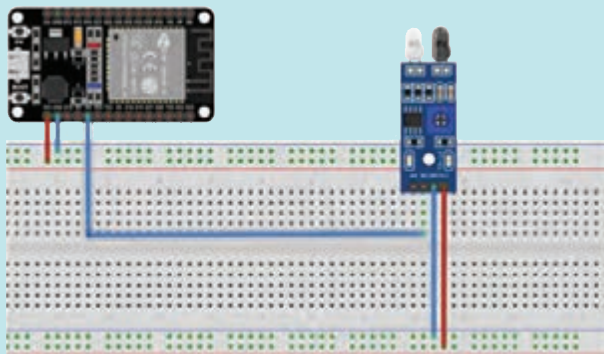
کار عملی ۶



مراحل اجرای کار:

۱ مطابق با شکل ۱۶، سخت افزار مورد نیاز را آماده کرده سپس با استفاده از توابع و دستورات، برنامه را بنویسید.

۲ مطابق با توضیحات ارائه شده در کار عملی ۱، سخت افزار را بصورت عملی و نرم افزاری در شبیه ساز Wokwi ببندید.



ب) مدار عملی روی برد

الف) شبیه سازی در wokwi

شکل ۱۶- نحوه اتصال ماژول IR به میکروکنترلر بصورت عملی و شبیه سازی در wokwi

۳ نرم افزار Arduino IDE را فعال نمایید و با زبان برنامه نویسی C برنامه زیر را بنویسید. با توجه به توضیح: دستورات، عملکرد هر خط از برنامه را در برنامه زیر بنویسید.

```
#define IR_sensor 4

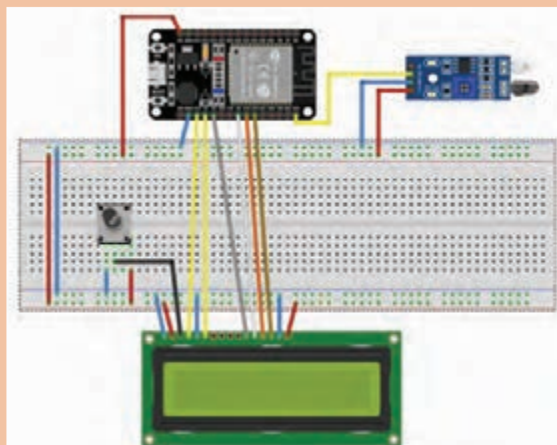
void setup() {
  Serial.begin(9600)
  pinMode(IR_sensor,INPUT) ;
  Serial.println("Reading IR Sensor...");}

void loop() {
  IR = digitalRead(IR_sensor);
  Serial.print("IR = ");
  Serial.println(IR);
  delay(1000);
}
```

- ۴ در صفحه وب [https://docs.wokwi.com/Guides/ESP32 Simulator](https://docs.wokwi.com/Guides/ESP32%20Simulator) وارد قسمت Guides/ESP32 Simulator شده و در قسمت Arduino Core قالب خام شبیه‌ساز (Arduino-ESP32 Project Template) را انتخاب نمایید. سپس مطابق با شکل ۱۷-الف، سخت‌افزار را بسته و با اجرای کد، شبیه‌سازی نمایید.
- ۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام نمایید. مدار را بر روی بردبرد بسته و پروژه را اجرا نمایید.
- ۶ عملکرد مدار را مشاهده کنید.
- ۷ نتایج حاصل از راه‌اندازی مدار را تحلیل کنید و به طور خلاصه بنویسید.

تمرین‌های زیر را با راهنمایی هنرآموز و به‌صورت گروهی طراحی و به‌صورت نرم‌افزاری (wokwi) و عملی اجرا نمایید.

۱ سیستم بسته‌بندی ساده: برنامه‌ای بنویسید که با استفاده از یک حسگر مادون قرمز (IR)، عبور محصولات از روی یک نوار نقاله را تشخیص داده و تعداد آنها را شمارش کند. تعداد محصولات شمارش شده باید به‌صورت لحظه‌ای روی LCD کاراکتری 2×16 نمایش داده شود. (شکل ۱۷)



ب



الف

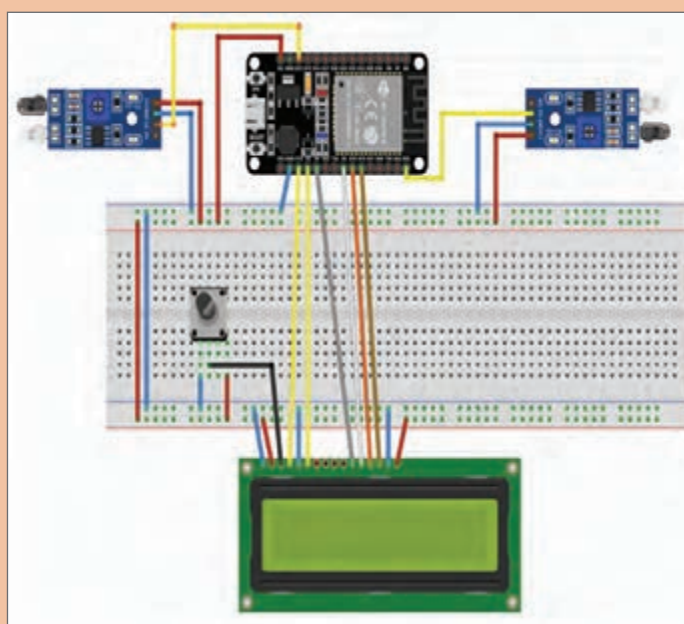
شکل ۱۷- مدار شبیه‌سازی سیستم بسته‌بندی ساده

- پس از رسیدن شمارش به عدد مشخص (مثلاً ۶)، پیام «Package Ready» روی LCD نمایش داده شود و شمارش از ۱ شروع شود.
- یک کلید برای ریست شمارنده به‌صورت دستی به طراحی اضافه شود.

۲ شبیه‌سازی سامانه کنترل ظرفیت پارکینگ (شکل ۱۸): برنامه‌ای بنویسید که مطابق با شکل ۱۹، با استفاده از دو حسگر مادون قرمز (IR)، عبور خودرو (ورود و خروج) از پارکینگ را تشخیص داده و ظرفیت باقیمانده پارکینگ به صورت لحظه‌ای روی LCD کاراکتری نمایش داده شود.



شکل ۱۸- کنترل ظرفیت پارکینگ با ESP32 و حسگرهای مادون قرمز (LR)



شکل ۱۹- شبیه‌سازی سیستم کنترل ظرفیت پارکینگ



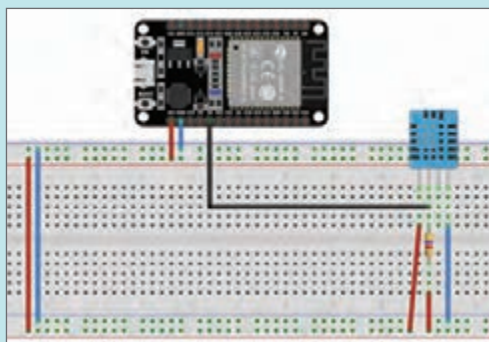
اتصال سنسور دما و رطوبت به میکروکنترلر

هدف پروژه: کسب مهارت در برنامه‌نویسی، شبیه‌سازی، پروگرام کردن میکروکنترلر و مدار سخت‌افزاری شامل سنسور DHT11 / DHT22

مواد، ابزار و تجهیزات: لوازم‌التحریر، دستورهای زبان C، برد توسعه ESP32 و برگه اطلاعات برد، رایانه دارای برنامه Arduino IDE، شبیه‌ساز Wokwi، LCD کاراکتری ۱۶×۲، پتانسیومتر ۱۰kΩ، بردبرد، جامپر و سنسور DHT11.

مراحل اجرای کار:

۱ مطابق با شکل ۲۰، سخت‌افزار موردنیاز را آماده کرده سپس مبتنی با توابع و دستورات، برنامه را بنویسید. توجه شود در صورت استفاده از ماژول سنسور، نیازی به مقاومت ۱۰k نیست.



شکل ۲۰

۲ مطابق با توضیحات ارائه شده در کار عملی ۱، کتابخانه DHT را جست‌وجو و در Arduino IDE و در شبیه‌ساز Wokwi نصب نمایید. دو تابع readHumidity() و readTemperature() در کتابخانه DHT، به ترتیب برای خواندن مقدار رطوبت و دما تعریف شده است. توجه شود که مقادیر دما و رطوبت از نوع اعداد اعشاری (float) می‌باشند.

۳ نرم‌افزار Arduino IDE را فعال نمایید و با زبان برنامه‌نویسی C برنامه زیر را بنویسید. با توجه به توضیح دستورات، عملکرد هر خط از برنامه را در شکل زیر بنویسید.

```
#include "DHT.h"
#define DHTTYPE DHT11
#define DHTPIN 4
DHT dht(DHTPIN, DHTTYPE);
void setup() {
  Serial.begin(9600)
  dht.begin();
  Serial.println("Reading DHT Sensor...");
}
void loop() {
  delay(2000);
  float humidity = dht.readHumidity();
  float temperature = dht.readTemperature();
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(temperature);
  Serial.println(" C");
}
```


۴ از مسیر زیر وارد محیط شبیه‌سازی wokwi شده

[/https://wokwi.com/Projects/new/ESP32](https://wokwi.com/Projects/new/ESP32)

سپس مطابق شکل ۲۰، سخت‌افزار را بسته و با اجرای کد، شبیه‌سازی نمایید.

۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام نمایید.

مدار را بر روی بردبرد بسته و پروژه را اجرا نمایید.

۶ عملکرد مدار را مشاهده کنید.

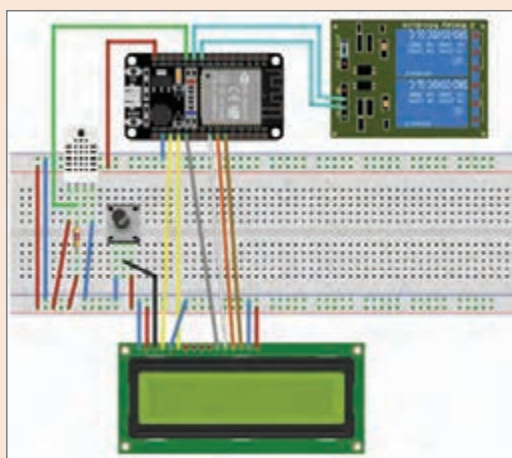
۷ نتایج حاصل از راه‌اندازی مدار را تحلیل کنید و به‌طور خلاصه بنویسید.

پروژه



تمرین‌های زیر را با راهنمایی هنرآموز و به‌صورت گروهی طراحی کرده و به‌صورت نرم‌افزاری (wokwi) و عملی اجرا نمایید.

۱ برنامه‌ای بنویسید که مقدار دما و رطوبت محیط توسط میکروکنترلر خوانده شده و بر روی نمایشگر LCD نمایش داده شود.



شکل ۲۱

۲ گلخانه هوشمند: برنامه‌ای بنویسید که مقدار دما و رطوبت مربوط به یک گلخانه توسط میکروکنترلر قرائت شود و بر روی نمایشگر LCD نمایش داده شود (شکل ۲۱). نمونه خروجی مورد انتظار مانند شکل ۲۲ می‌باشد.

□ اگر دما از ۲۰ درجه کمتر باشد، یک منبع گرمایی از طریق رله فعال شود و در خط اول LCD عبارت "Heat ON" نمایش داده شود و اگر دما از ۲۵ درجه بیشتر باشد، منبع گرمایی از طریق رله غیرفعال شود و در خط اول LCD عبارت "Heat OFF" نمایش داده شود.

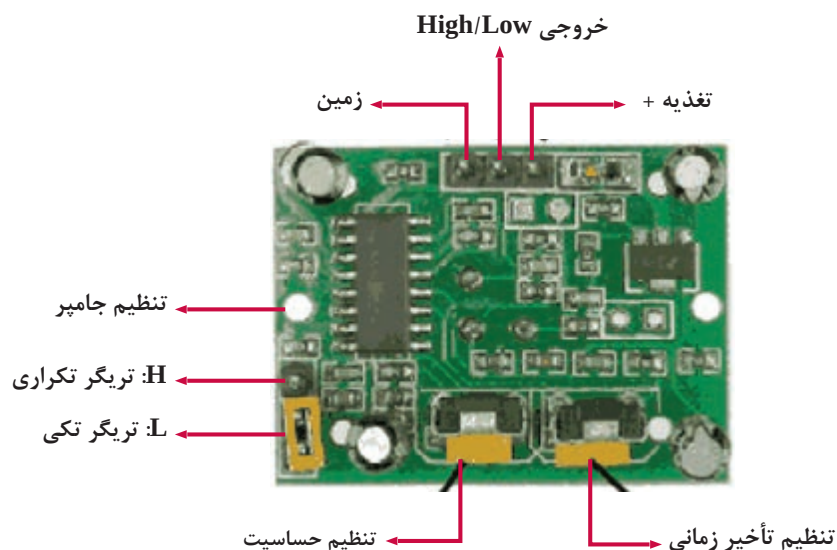
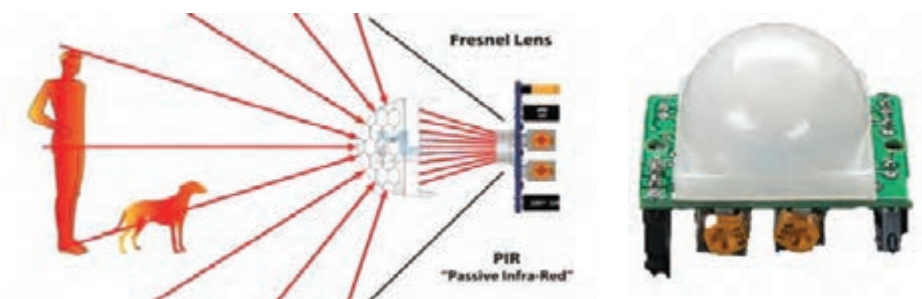


شکل ۲۲- خروجی گلخانه هوشمند

□ اگر رطوبت از ۷۵٪ درجه کمتر باشد، یک منبع رطوبت از طریق رله فعال شود و در خط دوم LCD عبارت "Hum ON" نمایش داده شود و اگر رطوبت از ۸۰٪ درجه بیشتر باشد، منبع رطوبت از طریق رله غیرفعال شود و در خط دوم LCD عبارت "Hum OFF" نمایش داده شود.

■ سنسور تشخیص حرکت (PIR:Passive Infra-Red)

PIR نوعی سنسور الکترونیکی می‌باشد که نور مادون قرمز تابش شده از اجسام را اندازه‌گیری می‌کند. این حسگر اغلب در حسگرهای حرکتی استفاده می‌شود. از موارد کاربرد این نوع حسگرها می‌توان به کاربردهای امنیتی، دوربین‌های فیلم‌برداری و برنامه‌های روشنایی خودکار اشاره کرد. از ویژگی‌های این حسگرها می‌توان به کوچک، ارزان و کم مصرف بودن و همچنین طول عمر بالا اشاره کرد. مطابق با شکل ۲۳، هنگامی که شخصی از کنار سنسور عبور می‌کند، تغییر سریع انرژی مادون قرمز تشخیص داده شده و یک سیگنال ارسال می‌شود.



شکل ۲۳

بر روی برد سنسورهای PIR دو پتانسیومتر برای تنظیم حساسیت و زمان تأخیر به کار رفته است.

- ۱ در مورد نحوه تنظیم حساسیت و زمان تأخیر سنسور تحقیق کرده و در کلاس مورد بحث قرار دهید.
- ۲ در مورد نحوه تریگر سنسور و تنظیم جامپر سنسور در شکل ۲۳، تحقیق کرده و در کلاس مورد بحث قرار دهید.

فعالیت





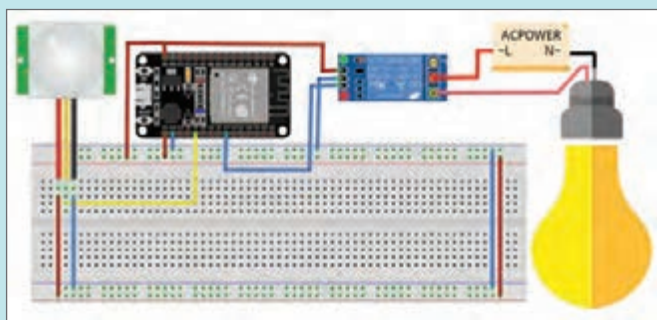
اتصال سنسور تشخیص حرکت به میکروکنترلر

هدف پروژه: کسب مهارت در برنامه‌نویسی، شبیه‌سازی، پروگرام کردن میکروکنترلر و مدار سخت‌افزاری شامل سنسور PIR

مواد، ابزار و تجهیزات: لوازم‌التحریر، دستوره‌های زبان C، برد توسعه ESP32 و برگه اطلاعات برد، رایانه دارای برنامه Arduino IDE، شبیه‌ساز Wokwi، LCD کاراکتری 16×2 ، پتانسیومتر $10k\Omega$ ، بردبرد، جامپر و سنسور PIR.

مراحل اجرای کار:

- ۱ مطابق با شکل ۲۴، سخت‌افزار مورد نیاز را آماده کرده سپس مبتنی با توابع و دستورات، برنامه را بنویسید.
- ۲ نرم‌افزار Arduino IDE را فعال نمایید و با زبان برنامه‌نویسی C برنامه زیر را بنویسید. با توجه به توضیح دستورات، عملکرد هر خط از برنامه را در شکل ۲۴ بنویسید.



با نظارت کامل هنرآموز کارگاه در اتصال ۲۲۰ ولت به مدار جهت روشن شدن لامپ دقت لازم به عمل آید.

نکات ایمنی



شکل ۲۴

```
#define lamp_Pin 18
#define PIR_Pin 4
int PIR
void setup() {
  Serial.begin(9600)
  pinMode(PIR_Pin, INPUT_PULLUP);
  delay(5000);}
void loop() {
  PIR = digitalRead(PIR_Pin);
  Serial.print("motion: ");
  Serial.println(PIR);
  If (PIR == 1)
  {
    digitalWrite(lamp_Pin.HIGH);
    Serial.print("motion is detected");
  }
  else
  {
    digitalWrite(lamp_Pin.LOW);
  }
}
```

۳ از مسیر زیر وارد محیط شبیه‌سازی wokwi شده

[/https://wokwi.com/Projects/new/ESP32](https://wokwi.com/Projects/new/ESP32)

سپس مطابق با شکل ۲۴، سخت‌افزار را بسته و با اجرای کد، شبیه‌سازی نمایید.

۴ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام نمایید.

مدار را بر روی بردبرد بسته و پروژه را اجرا نمایید.

۵ عملکرد مدار را مشاهده کنید.

۶ به ترتیب تأخیر زمانی و حساسیت سنسور را تغییر داده و عملکرد مدار را بررسی نمایید.

۷ نتایج حاصل از راه‌اندازی مدار را تحلیل کنید و به‌طور خلاصه بنویسید.

برنامه‌ای بنویسید که از طریق سنسور PIR حرکت تشخیص داده شود و سپس بر روی نمایشگر LCD پیام "welcome" به مدت ۵ ثانیه نمایش داده شود.

پروژه



خودآزمایی

۱ کدام گزینه، عملکرد اصلی پایه‌های GPIO در میکروکنترلر ESP32 را به‌درستی بیان می‌کند؟

الف فقط برای دریافت اطلاعات از حسگرها استفاده می‌شوند.

ب فقط برای ارسال فرمان به خروجی‌ها (مانند LED) به کار می‌روند.

پ هم برای ورودی (خواندن اطلاعات) و هم برای خروجی (ارسال فرمان) قابل استفاده هستند.

د وظیفه تأمین انرژی کل مدار را بر عهده دارند.

۲ برای اتصال یک «کلید فشاری» به ESP32، باید پایه مورد نظر را در چه حالتی پیکربندی کرد؟

الف خروجی دیجیتال (ب ورودی دیجیتال (پ آنالوگ (ت خاموش مطلق

۳ کدام یک از قطعات زیر، نمونه‌ای از یک «خروجی دیجیتال» محسوب می‌شود؟

الف کلید فشاری (Push Button) (ب سون‌سگمنت (پ سنسور دما (ت حسگر مادون قرمز

۴ در برنامه‌نویسی آردوینو برای ESP32، از کدام دستور برای تعیین وضعیت یک پایه به عنوان ورودی یا خروجی استفاده می‌شود؟

الف () digitalWrite (ب () digitalRead (پ () pinMode (ت () Serial.begin

۵ در یک سیستم کنترلی هوشمند، میکروکنترلر از طریق ورودی‌های دیجیتال، وضعیت محیط را دریافت کرده و پس از پردازش، از طریق خروجی‌ها پاسخ مناسب را اجرا می‌کند. □ صحیح □ غلط

۶ از پایه‌های دیجیتال می‌توان برای خواندن مقادیر پیوسته و متغیر (مانند تغییرات دمای محیط) به‌صورت مستقیم استفاده کرد.

□ درست □ نادرست

۷ دقت در سیم‌بندی و ترتیب اتصالات، تأثیری در عملکرد نهایی پروژه ندارد و فقط برنامه‌نویسی مهم است.

□ درست □ نادرست

۸ مفهوم GPIO را به صورت کوتاه بیان کنید.

۹ یک تفاوت اصلی بین ورودی دیجیتال و خروجی دیجیتال را نام ببرید.

۱۰ چرا در طراحی سیستم‌های کنترلی، داشتن «نگرش تحلیلی و مرحله‌به‌مرحله» ضروری است؟

ارزشیابی شایستگی ورودی‌ها و خروجی‌های دیجیتال در میکروکنترلر ESP32

نام و نام خانوادگی		شماره ملی	تاریخ ارزشیابی:	نوبت:
کد حرفه	۸۲۱۲۱۲	حرفه:	مونتازکاران تجهیزات و وسایل برقی و الکترونیکی	کد پیمان/پودمان
کد وظیفه شغل	۸۲۱۲۱۲۰۳	وظیفه شغل:	ساخت مدارهای دیجیتال و میکروکنترلر	سطح صلاحیت
کد کار	۸۲۱۲۱۲۰۳۵۱	کار:	برنامه نویسی پیشرفته (ESP32)	سطح شایستگی
			استاندارد عملکرد: تشریح ساختار کلی میکروکنترلر، انتخاب ESP32، یادگیری زبان برنامه‌نویسی و نوشتن برنامه ساده در محیط توسعه مرتبط	مهارت

ردیف	مراحل کار	شرایط انجام کار (ابزار، مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص‌ها/داوری/نمره دهی)	نتایج ممکن	نمره کسب شده
۱	تسلط بر بلوک دیاگرام و توانایی انتخاب فنی میکروکنترلر (مانند ESP32)	رایانه، برگه اطلاعاتی میکروکنترلرها، کاتالوگ قطعات	چک‌لیست عمل، مشاهده مستقیم فرایند نصب	تسلط کامل، انتخاب دقیق	۳	
				شناخت نسبی، انتخاب منطقی، نیاز به راهنمایی	۲	
				عدم شناخت، انتخاب نامناسب، ناتوانی نصب	۱	
۲	تسلط بر ویژگی‌های ساختاری و عملکردی میکروکنترلرها و بردهای توسعه و نحوه انتخاب یک میکروکنترلر مناسب با توجه به پروژه	رایانه، برگه اطلاعاتی میکروکنترلرها، کاتالوگ قطعات	چک‌لیست عمل، مشاهده مستقیم فرایند نصب	تسلط کامل، انتخاب دقیق	۳	
				شناخت نسبی، انتخاب منطقی، نیاز به راهنمایی	۲	
				عدم شناخت، انتخاب نامناسب، ناتوانی نصب	۱	
۳	بیان ویژگی‌های اصلی ESP32، انواع بردهای این خانواده و اجزای مهم یک برد توسعه ESP32	برگه اطلاعاتی چند نمونه از برد توسعه ESP32	چک‌لیست عمل، مشاهده مستقیم فرایند نصب	شناخت کامل، تفکیک دقیق، بیان دقیق	۳	
				شناخت کلی، برخی نقص‌ها، نیاز به راهنمایی	۲	
				عدم شناخت، اشتباهات فنی، ناتوانی بیان	۱	
۴	بیان انواع زبان‌های برنامه‌نویسی، شناخت تفاوت کامپایلر، مفسر و محیط توسعه	رایانه، نرم افزارهای Arduino ide، چند نمونه محیط توسعه و کامپایلرهای متداول	آزمون کتبی تشریحی، پرسش و پاسخ شفاهی فنی	تفکیک دقیق، شناخت کامل، بیان صحیح	۳	
				شناخت کلی، برخی ابهام‌ها، نیاز به راهنمایی	۲	
				عدم تشخیص، اشتباهات پایه، ناتوانی بیان	۱	
	شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش:	کارگاه الکترونیک یا سایت رایانه، تجهیزات کارگاهی و حفاظت فردی و دستور العمل‌های ایمنی و بهداشت حرفه‌ای	مشاهده رفتار هنرجو در حین انجام فعالیت، میزان رعایت اصول ایمنی و نظم، فعالیت‌های گروهی، مدیریت زمان و مسئولیت پذیری	رعایت کامل ایمنی، بهداشت و محیط زیست، رفتار حرفه‌ای، نظم و مسئولیت‌پذیری بالا	۲	
				رعایت نسبی ایمنی و بهداشت با خطاهای جزئی، رفتار کاری مناسب	۱	

☐ بلی

☐ خیر

ارزشیابی کار (شایستگی انجام کار)

توضیحات:

۱- معیار شایستگی انجام کار:

- برای احراز شایستگی، کسب حداقل نمره ۲ در هر یک از ۴ شاخص الزامی است (مراحل بحرانی).

- کسب حداقل نمره ۲ از بخش شایستگی‌های غیر فنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش

- کسب حداقل میانگین ۲ از مراحل کار

۲- تعریف سطوح شایستگی: صرف نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

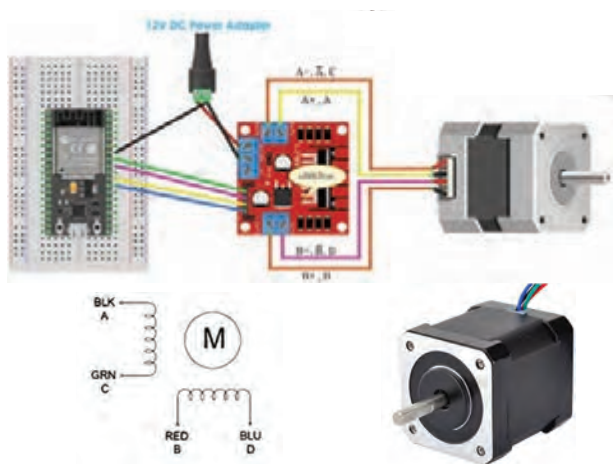
مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها

تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

واحد یادگیری ۲: ورودی و خروجی آنالوگ و کنترل موتور با ESP32

آیا تا به حال فکر کرده‌اید

- ۱ چگونه می‌توان کمیت‌های متغیر در دنیای واقعی مانند دما یا نور را که مقادیر پیوسته دارند، توسط میکروکنترلری که ماهیت دیجیتال دارد اندازه‌گیری کرد؟
 - ۲ چه تفاوت‌های اساسی در نحوه پردازش سیگنال‌های دیجیتال در مقایسه با سیگنال‌های آنالوگ وجود دارد؟
 - ۳ نقش اصلی مبدل آنالوگ به دیجیتال (ADC) در میکروکنترلر ESP32 چیست و چگونه داده‌های محیطی را به کدهای قابل فهم تبدیل می‌کند؟
 - ۴ برای تغییر پیوسته و تدریجی سرعت یک موتور DC یا شدت نور یک LED، چه تکنیک‌های کنترلی وجود دارد؟
 - ۵ طراحی سیستم‌های هوشمند چه ارتباطی با توانایی درک و تحلیل داده‌های آنالوگ توسط میکروکنترلر دارد؟
- میکروکنترلر ESP32، به‌عنوان ابزاری توانمند، امکان کنترل تجهیزاتی مانند LED، نمایشگرها و حسگرهای پایه را از طریق ورودی‌ها و خروجی‌های دیجیتال فراهم می‌کند. این ورودی‌ها و خروجی‌ها تنها دو وضعیت منطقی صفر و یک را تشخیص می‌دهند که برای کاربردهای مقدماتی مناسب است. با این وجود، بسیاری از کمیت‌های دنیای واقعی، نظیر شدت نور، دما، سرعت و میزان چرخش موتور، به‌صورت پیوسته تغییر می‌کنند. برای اندازه‌گیری و مدیریت چنین کمیت‌هایی، استفاده از ورودی‌ها و خروجی‌های دیجیتال به تنهایی کافی نبوده و به‌کارگیری سیگنال‌های آنالوگ ضروری است. در این واحد یادگیری، مفاهیم سیگنال‌های آنالوگ و سازوکار مبدل آنالوگ به دیجیتال (ADC) در ESP32 بررسی می‌شود. همچنین، نحوه دریافت مقادیر پیوسته ولتاژ و چگونگی به‌کارگیری درایورها برای کنترل موتورهای DC آموزش داده می‌شود. تلفیق ورودی‌های آنالوگ با روش‌های کنترلی، امکان طراحی سیستم‌های هوشمندی را فراهم می‌سازد که عملکرد آنها متناسب با تغییرات محیطی تغییر می‌کند. هدف نهایی، کسب مهارت در پیاده‌سازی کاربردهای واقعی میکروکنترلرها و آمادگی برای طراحی سیستم‌های کنترلی هوشمند است.

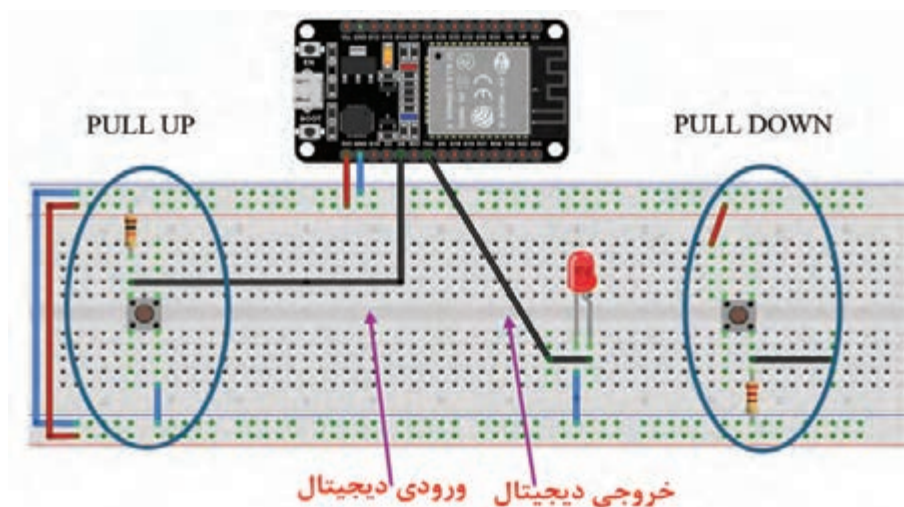


استاندارد عملکرد

تسلط بر پیکربندی و استفاده از پایه‌های آنالوگ ESP32 برای خواندن حسگرها و کنترل خروجی‌ها با دقت و نگرش سیستمی.

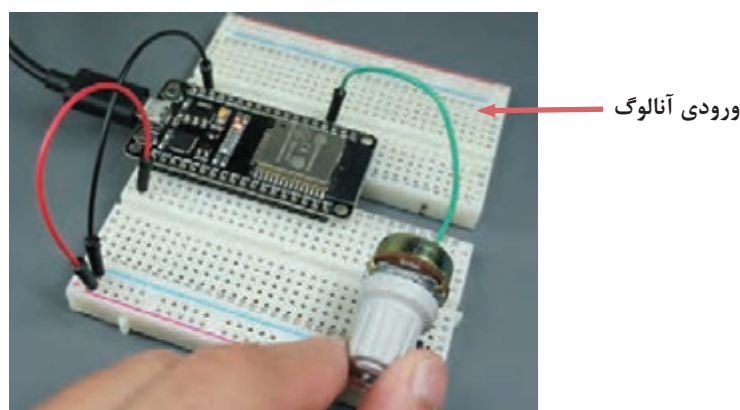
ورودی آنالوگ در میکروکنترلر

در درس‌های قبلی شما با ورودی‌ها و خروجی‌های دیجیتال آشنا شدید؛ یعنی پین‌هایی که مقدار آنها صفر (LOW) یا یک (HIGH) است. مطابق با شکل ۲۵، از میکروسوئیچ‌ها به عنوان ورودی دیجیتال ESP32 و از LED به عنوان خروجی دیجیتال استفاده شده است.



شکل ۲۵- ورودی و خروجی دیجیتال

این نوع سیگنال‌ها برای بسیاری از کاربردهای ساده مناسب‌اند، اما در دنیای واقعی کمیت‌ها اغلب پیوسته و قابل تغییر هستند، مانند شدت نور محیط، مقدار دما، ولتاژ خروجی یک سنسور، موقعیت ولوم و غیره. این کمیت‌ها آنالوگ نامیده می‌شوند؛ یعنی می‌توانند در یک بازه پیوسته تغییر کنند. برای اینکه یک میکروکنترلر مانند ESP32 بتواند چنین مقادیری را بخواند یا تولید کند، از ورودی و خروجی آنالوگ استفاده می‌کنیم. مطابق با شکل ۲۶، اگر یک ولوم را در مدار قرار داده و بچرخانیم، ولتاژ خروجی پتانسیومتر (پایه وسط) مقدار متغیری بین صفر تا $\frac{3}{3}$ ولت را خواهد داشت و مانند سوئیچ‌ها، فقط دو مقدار صفر و ۱ نیست.



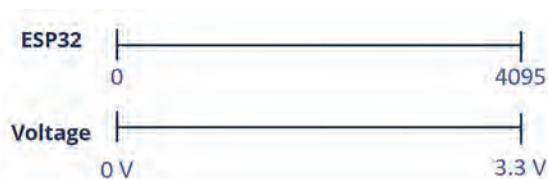
شکل ۲۶- ورودی آنالوگ



شکل ۲۷- عملکرد مبدل آنالوگ به دیجیتال در میکروکنترلر

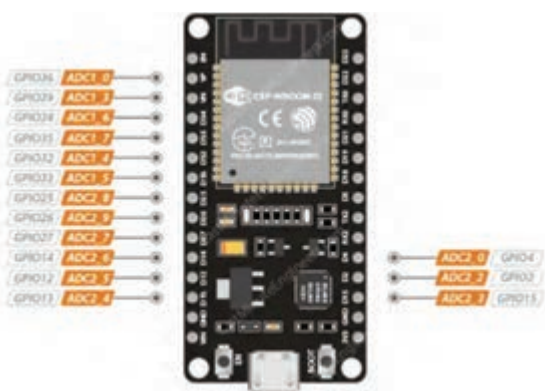
■ **مبدل ADC:** میکروکنترلرها در ذات خود فقط می‌توانند عددهای دیجیتال (صفر یا ۱) را پردازش کنند. پس برای اینکه مقدار یک سیگنال آنالوگ را بخوانند، باید آن را تبدیل به عدد دیجیتال کنند، مثلاً وقتی دمای محیط از ۲۰ به ۲۵ درجه افزایش پیدا می‌کند یا با تغییر ولوم، ولتاژ آنالوگ ورودی به صورت پیوسته (مثلاً از ۰/۶V به ۰/۷۵V) تغییر می‌کند، در میکروکنترلر مبدل آنالوگ به دیجیتال (ADC: Analog to Digital Converter)، ورودی آنالوگ را به یک عدد دیجیتال تبدیل می‌کند (شکل ۲۷).

■ **مفهوم رزولوشن (وضوح):** رزولوشن یعنی تعداد گام‌هایی که ADC می‌تواند تشخیص دهد. هرچه رزولوشن بالاتر باشد، یک میکروکنترلر می‌تواند تغییرات کوچک سیگنال آنالوگ ورودی را دقیق‌تر ببیند. پین‌های ADC ESP۳۲ به‌طور پیش فرض دارای رزولوشن ۱۲ بیتی هستند. این پین‌ها ولتاژ آنالوگ ورودی بین صفر تا ۳/۳ ولت را می‌خوانند و سپس عددی دیجیتال بین صفر تا ۴۰۹۵ (۲^{۱۲}) را بر می‌گردانند. لازم



شکل ۲۸- بازه سیگنال آنالوگ و دیجیتال

به ذکر است که رزولوشن قابل تغییر است. به عنوان مثال، ممکن است بخواهیم تنها وضوح ۱۰ بیتی داشته باشیم تا مقداری بین صفر تا ۱۰۲۳ به‌دست آوریم. در شکل ۲۸، بازه ولتاژ ورودی (آنالوگ) و بازه رزولوشن خروجی ADC (دیجیتال) برای رزولوشن ۱۲ بیتی نشان داده شده است.



شکل ۲۹- پین‌های ADC در برد توسعه ESP۳۲

■ **پین‌های آنالوگ به دیجیتال (ADC Pins) در ESP32:** مطابق شکل ۲۹، میکروکنترلر ESP32 از ۱۵ کانال (پین‌های فعال آنالوگ) پشتیبانی می‌کند که در دو طرف برد توسعه قرار دارند.

■ **دستور analogRead():** این دستور برای خواندن مقادیر آنالوگ ورودی مشابه با دستور digitalRead() که برای خواندن مقادیر دیجیتال ورودی معرفی شد، به کار می‌رود.



با دوستانتان در مورد سؤالات زیر بحث کنید و نتیجه را به کلاس ارائه دهید.

- وقتی ولوم را می چرخانیم، چه چیزی تغییر می کند؟
- ESP32 چگونه مقدار پیوسته ولوم را می فهمد؟
- چرا مقدار نمایش داده شده بین ۰ تا ۴۰۹۵ تغییر می کند؟



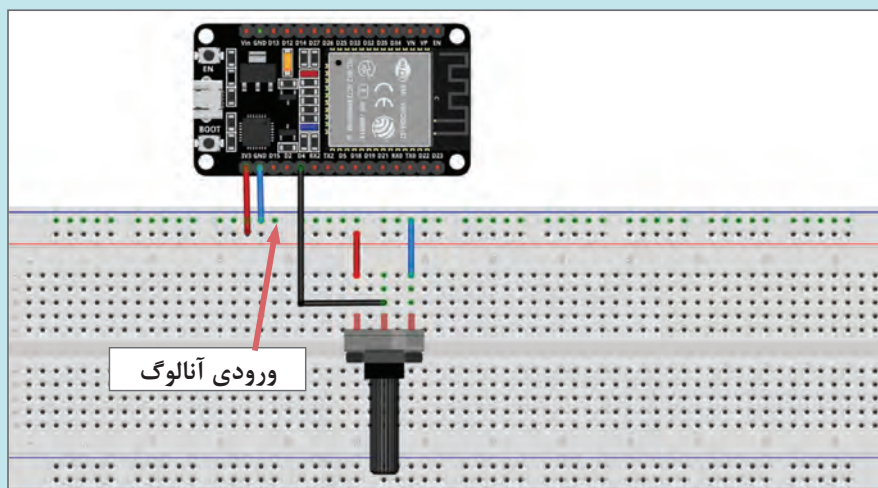
اتصال ولوم به میکروکنترلر ESP32

هدف پروژه: کسب مهارت در اتصال یک ولوم به میکروکنترلر ESP32 و برنامه ریزی، شبیه سازی و برنامه نویسی به جهت خواندن مقدار آنالوگ ورودی و نمایش مقدار خوانده شده به صورت عددی در Serial Monitor.

مواد، ابزار و تجهیزات: لوازم التحریر، دستورهای زبان C، برگه اطلاعات برد توسعه ESP32، برد توسعه ESP32، رایانه دارای Arduino IDE، شبیه ساز Wokwi، ولوم ۱۰k یک عدد، بردبرد، سیم جامپر، مقاومت.

مراحل اجرای کار:

- ۱ مطابق با شکل ۳۰، سخت افزار پروژه را به صورت عملی بر روی بردبرد ببندید. پایه ۴ به عنوان ورودی آنالوگ در نظر گرفته شده است. با چرخاندن ولوم، ولتاژ پایه وسط تغییر می کند. این ولتاژ عددی بین ۰ تا ۳/۳ ولت است. ADC این ولتاژ را به یک عدد دیجیتال بین ۰ تا ۴۰۹۵ تبدیل می کند.



شکل ۳۰- اتصال ولوم به میکروکنترلر ESP32

۲ نرم‌افزار Arduino IDE را فعال نمایید و با زبان برنامه‌نویسی C برنامه زیر را بنویسید.

```
int potPin = 4;    // analog pin define
int potValue = 0 ; // variable define
void setup() {
    Serial.begin(9600); }
void loop() {
    potValue = analogRead(potPin); // read analog pin
    Serial.print("Potentiometer Value: "); //
    Serial.println(potValue); //
    delay(500); // delay for ADC
}
```

۳ با توجه به دستورات آموزش داده شده، برنامه را تحلیل نمایید و در کلاس گزارش دهید.

۴ از مسیر زیر وارد محیط شبیه‌سازی wokwi شده

<https://wokwi.com/Projects/new/ESP32/>

سپس مطابق با شکل ۱۷-الف، سخت‌افزار را بسته و با اجرای کد، شبیه‌سازی نمایید.

۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام و پروژه را اجرا نمایید.

کار عملی ۲

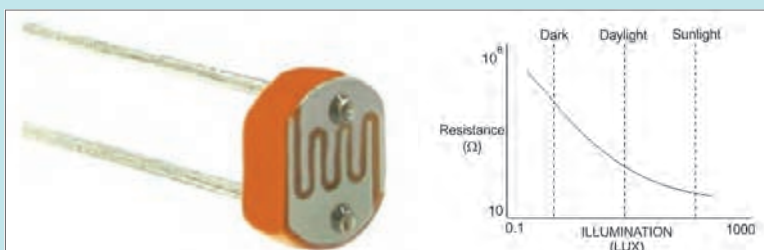


اتصال سنسور LDR به میکروکنترلر ESP32

هدف پروژه: تشخیص مقدار نور محیط توسط LDR به عنوان یک ورودی آنالوگ

مواد، ابزار و تجهیزات: لوازم‌التحریر، دستوره‌های زبان C، برگه اطلاعات برد توسعه ESP32، رایانه دارای Arduino IDE، شبیه‌ساز Wokwi، LDR، ولوم ۱۰k، LED، بردبرد، سیم جامپر، مقاومت.

■ **سنسور LDR:** Light Dependent Resistor یا نوعی مقاومت متغیر است که مقدار مقاومت آن به شدت نور محیط بستگی دارد. مطابق با شکل ۳۱، در نور زیاد، مقاومت LDR کم (در حد اهم) و در نور کم، مقاومت LDR زیاد (در حد مگا اهم) می‌باشد. به همین دلیل از LDR برای تشخیص شب و روز، شدت نور محیط، روشنایی اتاق‌ها و چراغ‌های هوشمند خیابانی استفاده می‌شود.



شکل ۳۱- برنامه خواندن مقدار نور محیط و نمایش آن

چرا LDR ورودی آنالوگ است؟ برخلاف کلید فشاری که فقط دو حالت دارد (صفر یا ۱)، LDR می‌تواند در شرایط نوری متفاوت، مشابه با ولوم، مقادیر پیوسته تولید کند. بنابراین خروجی LDR به صورت عدد آنالوگ توسط مبدل ADC میکروکنترلر خوانده می‌شود.

با استفاده از LDR چه پروژه‌هایی می‌توان ساخت؟ هنرجویان با مشورت گروهی حداقل ۳ ایده پیشنهاد دهند.

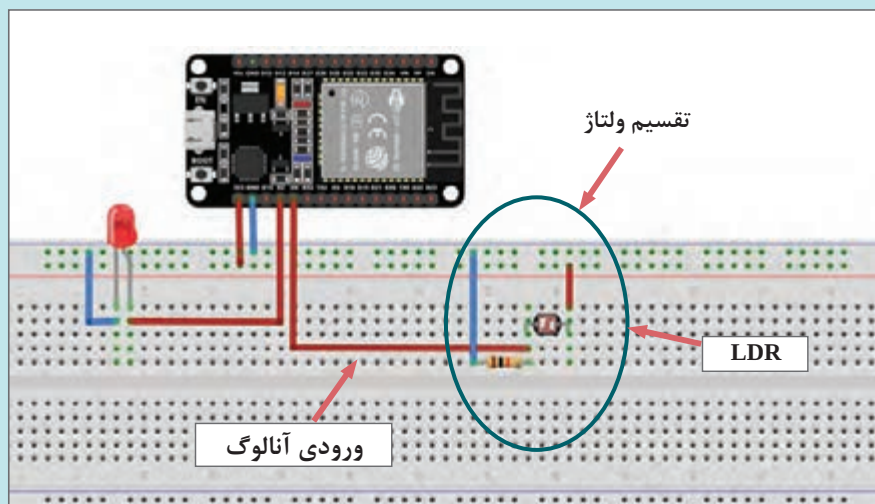
بارش فکری



مراحل اجرای کار:

۱ مطابق با مدار شکل ۸، دو مقاومت $10k$ و LDR با هم سری هستند و ولتاژ بین این دو تقسیم می‌شود. یک خروجی از بین این دو مقاومت به ESP32 متصل کرده‌ایم تا ولتاژ مقاومت $10k$ به ESP32 به عنوان یک مقدار آنالوگ وارد شود. مطابق با تحلیل رفتاری زیر، با تغییر نور محیط، مقدار مقاومت LDR تغییر می‌کند و باعث می‌شود ولتاژ ورودی به ESP32 تغییر کند (از صفر تا $3/3$ ولت).

$$\begin{array}{l} \text{Light} \uparrow \rightarrow R_{(LDR)} \downarrow \rightarrow V_{(LDR)} \downarrow \rightarrow V_{(10k)} \uparrow \\ \text{Light} \downarrow \rightarrow R_{(LDR)} \uparrow \rightarrow V_{(LDR)} \uparrow \rightarrow V_{(10k)} \downarrow \end{array}$$



شکل ۳۲- مدار عملی اتصال LDR به ESP32

۲ نرم‌افزار Arduino IDE را فعال نمایید و با زبان برنامه‌نویسی C برنامه زیر را بنویسید.

```
#define LDR_PIN 4  //
#define LED_PIN 2  //
void setup() {
  pinMode(LED_PIN, OUTPUT);  //
  Serial.begin(115200); }
void loop() {
  int lightValue = analogRead(LDR_PIN);  //
  Serial.println(lightValue);  //
  if (lightValue < 2000) {
    digitalWrite(LED_PIN, HIGH) ;//
  } else {
    digitalWrite(LED_PIN, LOW); }  //
  delay(300);}
```

۳ با توجه به دستورات آموزش داده شده، برنامه را تحلیل نمایید و در برنامه نوشته شده، جلوی خط‌های مشخص شده، توضیحات مناسب بنویسید.

۴ از مسیر زیر وارد محیط شبیه‌سازی wokwi شده

<https://wokwi.com/Projects/new/ESP32/>

سپس مطابق با شکل ۱۷-الف، سخت‌افزار را بسته و با اجرای کد، شبیه‌سازی نمایید.

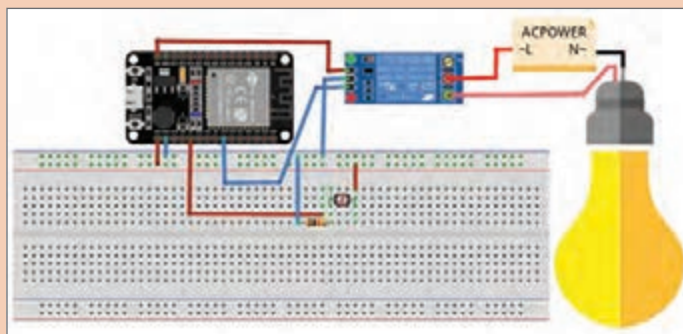
۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام و پروژه را اجرا نمایید.

تمرین



پروژه زیر را با راهنمایی هنرآموز به صورت نرم‌افزاری (wokwi) و عملی اجرا نمایید.

■ **پروژه فتوسل:** مطابق با شکل ۳۳، برنامه مدار کار عملی قبل (LDR)، را به گونه‌ای تغییر دهید که در زمان روشن بودن هوا، لامپ خاموش بوده و در Serial Monitor کلمه "روز" نوشته شود و هرگاه هوا تاریک شود، لامپ روشن شده و در Serial Monitor کلمه "شب" نوشته شود.



شکل ۳۳- برنامه فتوسل با ESP32



اتصال سنسور LM35 به میکروکنترلر ESP32

هدف پروژه: خواندن مقدار دمای محیط توسط سنسور دمای LM35 به عنوان یک ورودی آنالوگ و نمایش آن به صورت عددی در Serial Monitor.
مواد، ابزار و تجهیزات: لوازم التحریر، دستوره‌های زبان C، برگه اطلاعات برد توسعه ESP32، رایانه دارای Arduino IDE، شبیه‌ساز Wokwi، LM35، LED، برد بُرد، سیم جامپر، مقاومت.

بارش فکری



با استفاده از LM35 چه پروژه‌هایی می‌توان ساخت؟ با مشورت گروهی حداقل ۳ ایده پیشنهاد دهید.

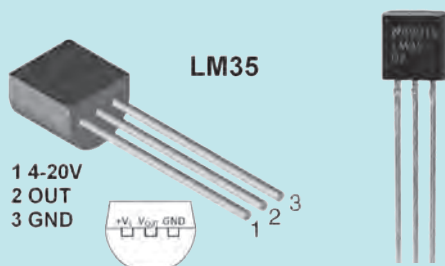
■ **سنسور دما:** یکی از ساده‌ترین و ارزان‌ترین راه‌ها برای افزودن قابلیت اندازه‌گیری دما به پروژه‌تان، استفاده از سنسور دمای LM35 است. LM35 یک سنسور دمای ولتاژ پایین است که می‌تواند دما را برحسب درجه سانتی‌گراد اندازه‌گیری کند. خروجی این سنسور، سیگنال ولتاژی است که به‌طور خطی با دما (برحسب درجه سانتی‌گراد) تغییر می‌کند، بنابراین استفاده از آن بسیار آسان است. سنسور دمای LM35 نسبتاً دقیق است، هرگز فرسوده نمی‌شود، در شرایط محیطی مختلف کار می‌کند و برای اندازه‌گیری دما به قطعات خارجی نیاز ندارد. علاوه بر این، سنسور LM35 نیازی به کالیبراسیون ندارد و معمولاً با دقت ± 0.5 درجه سانتی‌گراد در دمای اتاق و ± 1 درجه سانتی‌گراد در محدوده دمایی -55 تا $+155$ درجه سانتی‌گراد، دما را اندازه می‌گیرد. ولتاژ تغذیه این سنسور ۴ تا ۳۰ ولت و جریان مصرفی آن هنگام تبدیل دما کمتر از ۶۰ میکروآمپر است. همچنین خود گرمایی بسیار پایینی دارد (کمتر از 0.08 درجه سانتی‌گراد در هوای ساکن).

تحقیق کنید



در مورد پدیده خود گرمایی در قطعات الکترونیکی به‌ویژه سنسورها و دلیل ایجاد آن تحقیق کنید. خود گرمایی چه تأثیری بر عملکرد سنسور LM35 دارد و چگونه می‌توان آن را کاهش داد؟

■ **پایه‌های سنسور LM35:** LM35 به شکل‌های مختلف عرضه می‌شود، اما رایج‌ترین نوع آن مطابق شکل ۳۴، پکیج سه پایه TO-۹۲ است که دقیقاً شبیه یک ترانزیستور است.



شکل ۳۴- سنسور دمای LM35

پایه ۲، یک ولتاژ آنالوگ تولید می‌کند که با دما رابطه مستقیم دارد و باید به پایه ورودی آنالوگ (ADC) متصل شود.

■ **خواندن داده‌های دمای آنالوگ: LM35** دمای محیط را به صورت یک ولتاژ آنالوگ در خروجی خود (پایه ۲) آشکار می‌سازد و مطابق توضیحات مربوط به ADC، این ولتاژ در میکروکنترلر به صورت یک عدد در بازه صفر تا ۴۰۹۵ تبدیل می‌شود. بنابراین در برنامه‌نویسی به صورت روند برعکس، ابتدا باید عدد تولید شده را به ولتاژ و سپس از ولتاژ به دما تبدیل کرد.

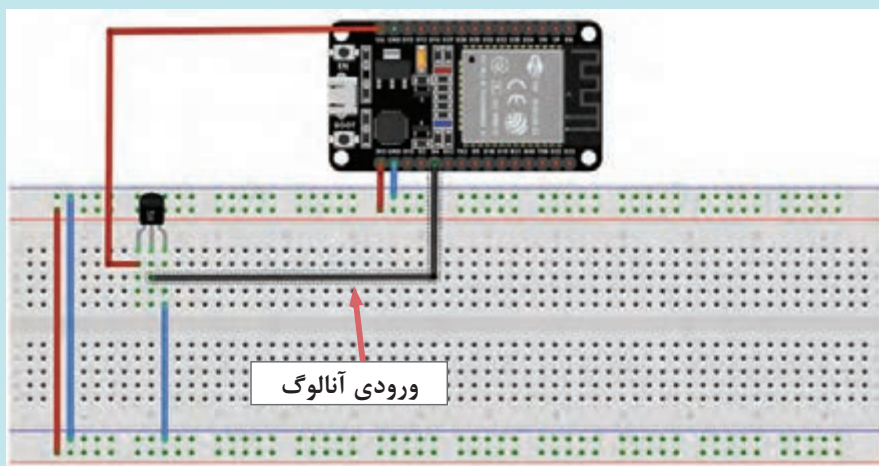
$$\text{ولتاژ خروجی} = (۴۰۹۵ / ۵) \times (\text{مقدار خوانده شده از سنسور})$$

این فرمول عدد صفر تا ۴۰۹۵ را به صفر تا ۵ ولت تبدیل می‌کند. سپس برای تبدیل ولت به دما از فرمول زیر استفاده می‌شود:

$$\text{دما (برحسب درجه سانتی‌گراد)} = \text{ولتاژ خروجی} \times ۱۰۰$$

مراحل اجرای کار:

۱ مطابق با مدار شکل ۳۵، برای تغذیه سنسور از ولتاژ ۵ ولت ESP32 (Vin) استفاده می‌کنیم تا محدوده ولتاژ مورد نیاز سنسور فراهم شود.



شکل ۳۵- مدار عملی اتصال LM35 به ESP32

۲ نرم افزار Arduino IDE را فعال نمایید و با زبان برنامه نویسی C برنامه زیر را بنویسید.

```
#define LM35_PIN 4 //
void setup() {
  Serial.begin (9600) ;
  pinMode (LM35_PIN, INPUT) ; //
}

void loop() {
  float temp = analogRead (LM35_PIN) ; //
  temp = temp * (5/4096) ; //
  temp = temp * (100) ; //
  Serial.print ("TEMPERATURE = ") ; //
  Serial.print (temp); //
  Serial.println ("c"); //
  delay(1000);
}
```

۳ با توجه به دستورات آموزش داده شده، برنامه را تحلیل نمایید و برنامه نوشته شده، جلوی خط‌های مشخص شده، توضیحات مناسب بنویسید و در کلاس گزارش دهید.

۴ در صفحه وب [https://docs.wokwi.com](https://docs.wokwi.com/Guides/ESP32%20Simulator) و قسمت Guides/ESP32 Simulator شده و در قسمت Arduino Core قالب خام شبیه ساز (Arduino_ESP32 Project Template) را انتخاب نمایید. سپس مدار را در شبیه ساز بسته و اجرا نمایید.

۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام و پروژه را اجرا نمایید.

تمرین



پروژه زیر را با راهنمایی هنرآموز به صورت نرم افزاری (wokwi) و عملی اجرا نمایید.

- کار عملی ۳ را تکرار کنید با این تفاوت که مقدار دما را بر روی LCD نمایش دهید.
- کار عملی ۳ را به این صورت تکمیل کنید که اگر دما از ۳۰ درجه بیشتر شود یک LED روشن شود.

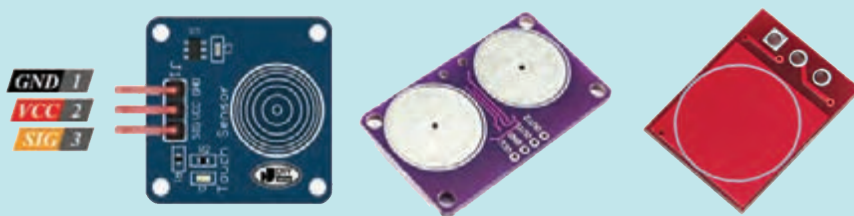
کار عملی ۴



اتصال سنسور لمسی (Touch) به میکروکنترلر ESP32

هدف پروژه: خواندن مقدار سنسور لمسی به عنوان یک ورودی آنالوگ و نمایش آن به صورت عددی در Serial Monitor.

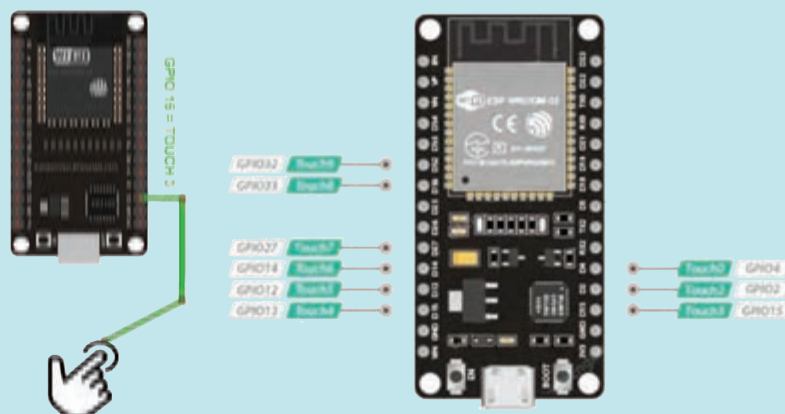
مواد، ابزار و تجهیزات: لوازم التحریر، دستوره‌های زبان C، برگه اطلاعات برد توسعه ESP32، رایانه دارای Arduino IDE، شبیه‌ساز Wokwi، سنسور لمسی، LED، برد برد، سیم جامپر، مقاومت. در بسیاری از دستگاه‌های امروزی مانند تلفن‌های همراه، پنل‌های لمسی، لامپ‌های هوشمند و تجهیزات صنعتی جدید، به جای کلیدهای فشاری مکانیکی از ورودی‌های لمسی استفاده می‌شود. این نوع ورودی‌ها باعث افزایش طول عمر سیستم، زیبایی ظاهری و سهولت استفاده می‌شوند. میکروکنترلر ESP32 به‌صورت داخلی دارای قابلیت تشخیص لمس (Touch Sensing) است. این ویژگی باعث می‌شود بدون نیاز به قطعات اضافی، بتوان ورودی لمسی را در پروژه‌های مختلف پیاده‌سازی کرد.



شکل ۳۶- انواع ماژول از سنسور لمسی

■ **طرز کار ورودی‌های لمسی:** ورودی‌های لمسی در ESP32 بر اساس تغییر ظرفیت خازنی کار می‌کنند. بدن انسان دارای بار الکتریکی بسیار ضعیفی است و هنگامی که انگشت دست به یک پایه لمسی متصل به ESP32 نزدیک یا آن را لمس می‌کند، ظرفیت الکتریکی آن پایه تغییر می‌کند. میکروکنترلر این تغییر را تشخیص داده و به‌صورت یک مقدار عددی در اختیار برنامه قرار می‌دهد. هرچه لمس قوی‌تر یا سطح تماس بیشتر باشد، مقدار خوانده‌شده تغییر بیشتری خواهد داشت. سنسورهای لمسی به‌صورت ماژول نیز در دسترس می‌باشد که در شکل ۳۶ چند نمونه از آنها نشان داده شده است.

■ **پایه‌های لمسی در ESP32:** همه پایه‌های ESP32 قابلیت لمس ندارند، بنابراین باید حتماً از پایه‌های مخصوص Touch استفاده شود. ESP32 دارای چندین پایه مخصوص لمس است که با نام Touch Pins شناخته می‌شوند. در شکل ۳۷، این پایه‌ها مشخص شده است.



شکل ۳۷- پایه‌های لمسی در ESP32

برای خواندن مقدار لمس از پایه‌های لمسی ESP32، از دستور زیر استفاده می‌شود:

```
touchRead(pin);
```

pin شماره پایه لمسی است و خروجی این دستور یک عدد صحیح است که مقدار آن با لمس کردن پایه کاهش پیدا می‌کند.

مراحل اجرای کار:

۱ مطابق با مدار شکل ۳۷، پایه ۱۵ را به عنوان ورودی لمسی در نظر بگیرید. مطابق با شکل ۳۸، مقدار سنسور قبل و بعد از لمس تغییر می‌کند.

قبل از لمس	بعد از لمس
Output Serial Monitor X	Output Serial Monitor X
Message (Enter to send message to 'ES	Message (Enter to send message to 'ES
75	14
75	13
75	13
74	13
75	13
74	12
75	12
74	14
	15

شکل ۳۸- مقدار پایه لمسی در ESP32 قبل و بعد از لمس

۲ نرم‌افزار Arduino IDE را فعال نمایید و با زبان برنامه‌نویسی C برنامه زیر را بنویسید.

```
void setup() {
    Serial.begin(9600); //
}
void loop() {
    int touchValue = touchRead(4); //
    Serial.println(touchValue); //
    delay(300);
}
```

۳ با توجه به دستورات آموزش داده شده، برنامه را تحلیل نمایید و در برنامه نوشته شده، جلوی خط‌های مشخص شده، توضیحات مناسب بنویسید و در کلاس گزارش دهید.

۴ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام و پروژه را اجرا نمایید.

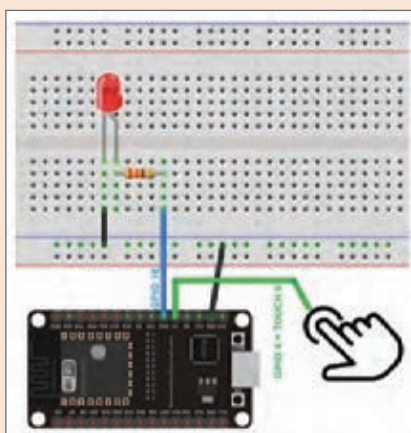
بررسی کنید



تغییر اندازه سطح فلزی متصل به پایه لمسی چه تأثیری بر حساسیت لمس دارد. برای بررسی این موضوع می‌توانید از یک فویل آلومینیومی مطابق با شکل ۳۹ استفاده نمایید.

شکل ۳۹- سنسور لمسی با فویل آلومینیومی

پروژه



پروژه زیر را با راهنمایی هنرآموز به صورت عملی اجرا نمایید. مطابق با شکل ۴۰، برنامه‌ای بنویسید که با لمس یک پایه لمسی در ESP32، یک LED روشن شود و با لمس مجدد همان پایه، LED خاموش گردد. برای تشخیص لمس، یک آستانه مناسب (Threshold) تعیین کنید.

شکل ۴۰- کنترل LED با سنسور لمسی

خروجی آنالوگ در میکروکنترلر

در بخش قبل با ورودی‌های آنالوگ آشنا شدیم و دیدیم که چگونه می‌توان اطلاعات پیوسته‌ای مانند ولتاژ یا شدت نور را از محیط دریافت و به صورت عددی در میکروکنترلر پردازش کرد. اما دریافت اطلاعات تنها نیمی از یک سیستم کنترلی است. در بسیاری از کاربردهای واقعی، لازم است میکروکنترلر بتواند خروجی‌هایی تولید کند که مقدار آنها به صورت تدریجی و قابل تنظیم تغییر کند؛ برای مثال کم و زیاد شدن نور یک چراغ، تغییر سرعت یک موتور یا تنظیم شدت صدا.

کار عملی ۵



تغییر تدریجی نور LED (LED Fade)

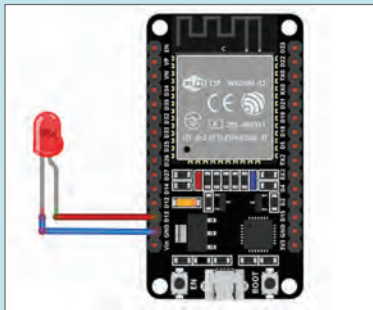
هدف پروژه: تغییر تدریجی نور یک LED به صورت کم و زیاد به جهت درک مفهوم خروجی آنالوگ و نمایش آن به صورت عددی در Serial Monitor.

مواد، ابزار و تجهیزات: لوازم التحریر، دستورهای زبان C، برگه اطلاعات برد توسعه ESP32، رایانه دارای Arduino IDE، شبیه‌ساز Wokwi، LED، برد برد، سیم جامپر، مقاومت.



■ **دستور `analogWrite()`:** در بخش‌های قبلی، برای کنترل خروجی‌ها از دستور `digitalWrite` استفاده کردیم. این دستور یک مقدار دیجیتال که فقط دو حالت دارد روشن (HIGH) و خاموش (LOW) در پایه خروجی ایجاد می‌کند. اما در بسیاری از کاربردهای واقعی، لازم است مقدار خروجی به صورت تدریجی (آنالوگ) تغییر کند؛ برای مثال کم و زیاد شدن نور یک LED یا تنظیم سرعت یک موتور. برای انجام چنین کارهایی از دستور `analogWrite` استفاده می‌شود. این دستور یک مقدار عددی (i) را به پایه خروجی (LED_pin) ارسال می‌کند.

`analogWrite(LED_PIN, i);`



مراحل اجرای کار:

۱ مطابق با مدار شکل ۴۱، پایه ۱۳ را به عنوان خروجی آنالوگ در نظر بگیرید و به یک LED متصل نمایید.

شکل ۴۱- کنترل LED به صورت آنالوگ

۲ نرم‌افزار Arduino IDE را فعال نمایید و با زبان برنامه‌نویسی C برنامه زیر را بنویسید.

```
define LED_PIN 13 //
void setup() {
    pinMode(LED_PIN, OUTPUT); //
}
void loop() {
    // افزایش تدریجی نور
    for (int i = 0; i <= 255; i++) //
    {
        analogWrite(LED_PIN, i); //
        delay(50); }
    // کاهش تدریجی
    for (int i = 255; i >= 0; i--) //
    {
        analogWrite(LED_PIN, i); //
        delay(50); }
}
```

۳ با توجه به دستورات آموزش داده شده، برنامه را تحلیل نمایید و جلوی خط‌های مشخص شده، توضیحات مناسب بنویسید و در کلاس گزارش دهید.

در این پروژه، با استفاده از دو حلقه، به ترتیب مقدار خروجی افزایش و سپس کاهش می‌یابد. با افزایش مقدار خروجی، نور LED بیشتر و با کاهش آن، نور LED کاهش می‌یابد. در نتیجه نور LED به صورت نرم و تدریجی پر نور و کم نور می‌شود. مقدار تأخیر (delay) سرعت تغییر نور را مشخص می‌کند به طوری که هرچه مقدار تأخیر کمتر باشد، تغییر نور سریع‌تر انجام می‌شود و بالعکس.

۳ از مسیر زیر وارد محیط شبیه‌سازی wokwi شده

<https://wokwi.com/Projects/new/ESP32/>

سپس مدار را در شبیه‌ساز بسته و اجرا نمایید.

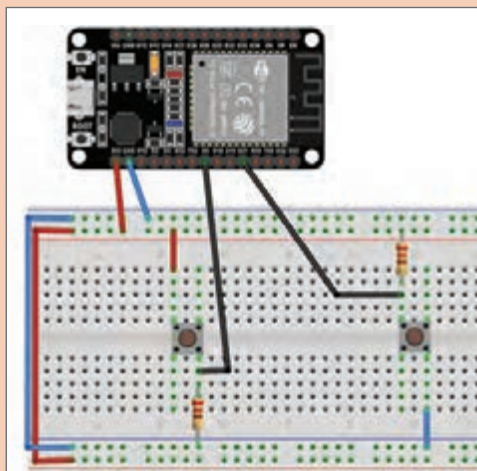
۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام و پروژه را اجرا نمایید.

تمرین



پروژه زیر را با راهنمایی هنرآموز و به صورت گروهی به صورت نرم‌افزاری (wokwi) و عملی اجرا نمایید.

مطابق با شکل ۴۲، برنامه‌ای بنویسید که با فشردن یک کلید به عنوان ورودی دیجیتال در ESP32، نور یک LED از کم به زیاد تغییر یابد و با کلید دیگر کاهش یابد. (در مدار شکل ۴۲، از LED روی برد توسعه استفاده شده است.)



شکل ۴۲- مدار کنترل نور LED به صورت آنالوگ توسط دو کلید

کار عملی ۶



کنترل شدت نور LED با ولوم (Potentiometer)

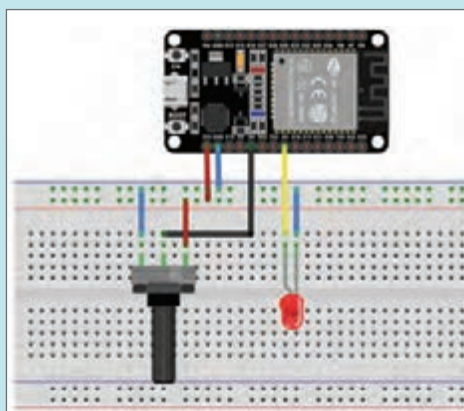
هدف پروژه: کنترل شدت نور یک LED با استفاده از یک ولوم به جهت درک مفهوم ورودی و خروجی آنالوگ و نمایش آن به صورت عددی در Serial Monitor.

مواد، ابزار و تجهیزات: لوازم التحریر، دستورهای زبان C، برگه اطلاعات برد توسعه ESP32، رایانه دارای Arduino IDE، شبیه‌ساز Wokwi، LED، ولوم ۱۰k، بردبرد، سیم جامپر، مقاومت.

در پروژه‌های واقعی، معمولاً مقدار یک ورودی آنالوگ (مانند ولوم، نور یا دما) مستقیماً برای کنترل یک خروجی استفاده می‌شود. هدف از این پروژه، کنترل یک خروجی آنالوگ با استفاده از یک ورودی آنالوگ می‌باشد.

مطابق با پروژه‌های قبل، ورودی آنالوگ در بازه ولتاژ ۰ تا ۳/۳۷ متغیر است و پس از پردازش توسط مبدل آنالوگ به دیجیتال (ADC) در میکروکنترلر ESP۳۲، به یک مقدار عددی در بازه ۰ تا ۴۰۹۵ (رزولوشن ۱۲ بیت) تبدیل می‌شود. از طرفی دیگر، خروجی آنالوگ شبیه‌سازی شده با دستور analogWrite در بازه ۰ تا ۲۵۵ تغییر می‌کند. بنابراین برای ارتباط دادن ورودی آنالوگ به خروجی، لازم است مقدار خوانده شده از ورودی، متناسب‌سازی شده و از بازه ۰-۴۰۹۵ به بازه ۰-۲۵۵ تبدیل گردد. این متناسب‌سازی مطابق با رابطه زیر انجام می‌شود.

$$۴۰۹۵ \div ۱۶ = ۲۵۵$$



مراحل اجرای کار:

۱ مطابق با مدار شکل ۴۳، پایه ۴ را به عنوان ورودی آنالوگ و پایه ۵ را به عنوان خروجی آنالوگ در نظر بگیرید و به ترتیب به یک ولوم و LED متصل نمایید.

شکل ۴۳- کنترل شدت نور LED
به صورت آنالوگ توسط ولوم

۲ نرم‌افزار Arduino IDE را فعال نمایید و با زبان برنامه‌نویسی C برنامه زیر را بنویسید.

```
int potPin = 4; //
int ledPin = 5; //
void setup() {
    pinMode(ledPin, OUTPUT); //
}
void loop() {
    int potValue = analogRead(potPin); // خواندن ولوم
    int LEDValue = potValue//16 ; // تبدیل بازه ورودی آنالوگ به خروجی آنالوگ
    analogWrite(ledPin, LEDValue); // LED کنترل نور
}
```

۳ با توجه به دستورات آموزش داده شده، برنامه را تحلیل نمایید و در مقابل خط‌های مشخص شده، توضیحات مناسب بنویسید.

در این پروژه، با استفاده از دو دستور analogRead() و analogWrite()، مقدار ورودی آنالوگ (ولوم) خوانده شده و به خروجی آنالوگ (LED) منتقل می‌شود.

۴ از مسیر زیر وارد محیط شبیه‌سازی wokwi شده

<https://wokwi.com/Projects/new/ESP32/>

سپس مدار را در شبیه‌ساز بسته و اجرا نمایید.

۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام و پروژه را اجرا نمایید.

کنترل موتور DC با میکروکنترلر

موتور الکتریکی نوعی ماشین است که توانایی تبدیل انرژی الکتریکی به انرژی مکانیکی را دارد. در یک موتور الکتریکی در اثر عبور جریان الکتریسیته از داخل سیم پیچ، میدان مغناطیسی تولید و باعث دوران شافت می‌شود.

■ **موتور DC:** نوعی موتور الکتریکی است که با جریان مستقیم یا DC کار می‌کند. سرعت موتور DC وابسته به ولتاژ و گشتاور موتور DC وابسته به جریان ورودی است. از انواع موتور DC مطابق با شکل ۴۴، می‌توان به آرمیچر، استپر، سروو و سروو موتور اشاره کرد.



آرمیچر



استپر



سروو

شکل ۴۴- انواع موتور DC

■ **آرمیچر:** یک موتور الکتریکی ساده با قابلیت کار با جریان مستقیم DC است. دارای سیم پیچ داخلی و شافت متحرک است. با اعمال ولتاژ به موتور DC شافت آن دور می‌گیرد. در شکل ۴۵، مدل‌های گیربکس‌دار و ساده آن نشان داده شده است.

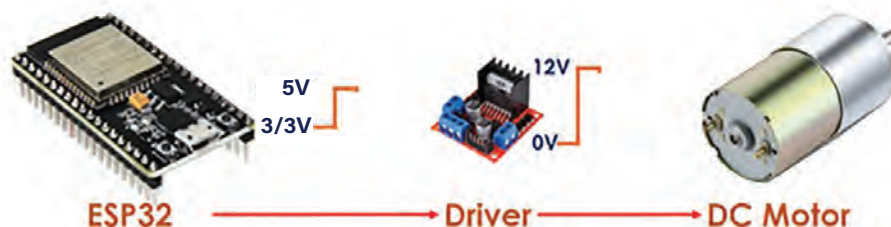


80RPM

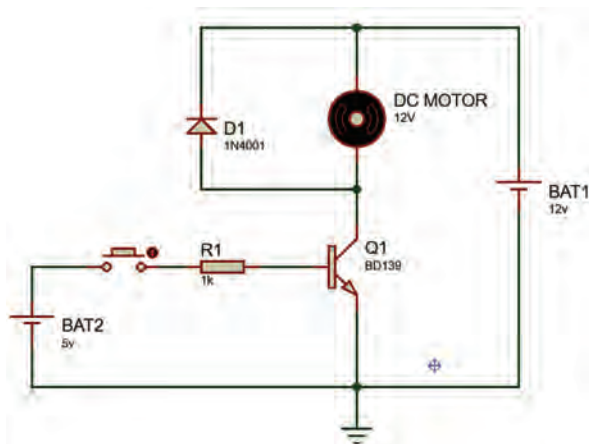


شکل ۴۵- انواع آرمیچر

■ **درایور موتور:** اتصال مستقیم موتور به میکروکنترلر به این علت که موتورهای نیاز به جریان بالا برای راه‌اندازی و کار کردن دارند، صحیح نیست و باعث آسیب دیدن میکروکنترلر می‌شود و بنابراین نیاز به یک واسط بین میکروکنترلر و موتور می‌باشد که از طریق میکروکنترلر فرمان بگیرد و در عین حال تأمین‌کننده جریان مورد نیاز موتور از مسیری جدای از میکروکنترلر باشد. این واسط، درایور (DRIVER) نام دارد (شکل ۴۶). درایورها در اصل مدارهای ترانزیستوری هستند که نقش سوئیچ و تأمین جریان موتور را بر عهده دارند. در شکل ۴۷، یک مدار درایور ساده ترانزیستوری نشان داده شده است.



شکل ۴۶- نقش درایور در راه‌اندازی موتور

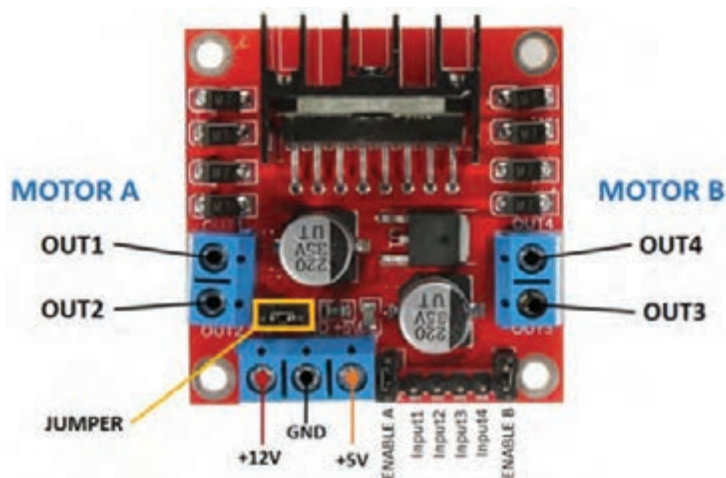


شکل ۴۷- یک درایور ساده ترانزیستوری

امروزه درایورها به صورت ماژول‌های آماده در انواع مختلف موجود می‌باشد که در ادامه یکی از معروف‌ترین درایورها معرفی می‌شود.

■ **درایور L۲۹۸N:** درایور موتور L۲۹۸N برای اکثر موتورهای سرگرمی که برای کار به ولتاژ ۶ ولت یا ۱۲ ولت نیاز دارند مناسب است. درایور موتور L۲۹۸N می‌تواند تا ۳ آمپر - ۳۵ ولت را مدیریت کند. علاوه بر این، به ما اجازه می‌دهد تا دو موتور DC را به طور هم‌زمان کنترل کنیم که برای ساخت یک ربات عالی است. درایور موتور L۲۹۸N در شکل ۴۸ نشان داده شده است.

این درایور دارای دو بلوک ترمینال در هر طرف برای هر موتور است OUT۱ و OUT۲ در سمت چپ و OUT۳ و OUT۴ در سمت راست. خروجی OUT۱ و OUT۳ مربوط به ترمینال + موتورهای و خروجی OUT۲ و OUT۴ مربوط به ترمینال - موتورهای می‌باشند. در قست پایین، یک بلوک سه ترمیناله وجود دارد که ۱۲V، GND، و ۵V+ برای تأمین ولتاژ تراشه L۲۹۸N استفاده می‌شود.



شکل ۴۸- درایور L۲۹۸

توجه شود که اگر جامپر در جای خود قرار داشته باشد، تراشه با استفاده از منبع تغذیه موتور تغذیه می‌شود و نیازی به تأمین ۵ ولت از طریق ترمینال ۵+ ولت نیست و به عنوان یک خروجی ۵ ولت عمل می‌کند. همچنین اگر تغذیه موتور بیش از ۱۲ ولت باشد، باید جامپر برداشته شود.

در قسمت پایین - سمت راست ماژول چهار پین ورودی ($IN1, IN2, IN3, IN4$) و دو پایه فعال‌ساز ($EN1, EN2$) وجود دارد. پایه‌های ورودی برای کنترل جهت موتورهای DC و پین‌های فعال‌ساز برای کنترل هر موتور استفاده می‌شود. $IN1$ و $IN2$ و $EN1$ مربوط به موتور A و $IN3$ و $IN4$ و $EN2$ مربوط به موتور B می‌باشند.

روی پین‌های فعال‌ساز به‌طور پیش فرض جامپر قرار داده شده است. برای کنترل سرعت موتورها، جامپر باید برداشته شود. در جدول زیر نحوه عملکرد پین‌های فعال‌ساز توضیح داده شده است:

جدول ۳

حالت موتور	ENABLE پایه (EN)
موتور فعال	HIGH (۱)
موتور غیر فعال	LOW (۰)

برای تغییر جهت موتور، نحوه اعمال ولتاژ به پایه‌ها را مطابق شکل ۴۹ باید تغییر داد.



شکل ۴۹- نحوه تغییر جهت چرخش موتور

در جدول ۴ نحوه عملکرد پین‌های ورودی (IN۱, IN۲, IN۳, IN۴) توضیح داده شده است.

جدول ۴

IN۱(IN۳)	IN۲(IN۴)	جهت موتور
HIGH	LOW	موتور A (B) ساعتگرد
LOW	HIGH	موتور A (B) پاد ساعتگرد
HIGH	HIGH	موتور A (B) متوقف
LOW	LOW	موتور A (B) متوقف

در مورد انواع مختلف درایور موتور DC تحقیق کرده و با درایور L۲۹۸ مقایسه نمایید.

تحقیق کنید



کار عملی ۷



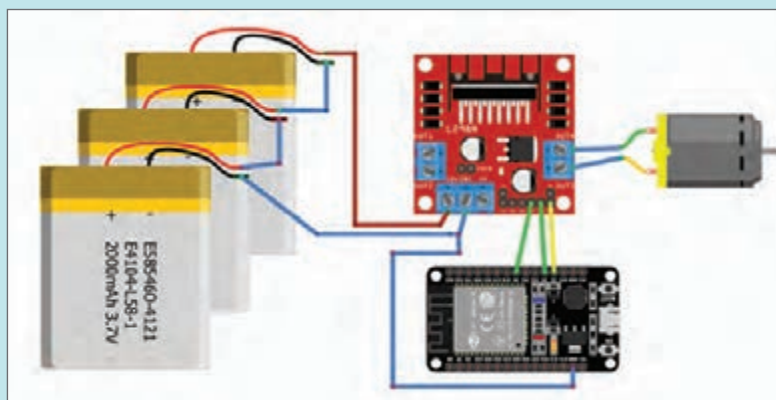
کنترل موتور DC مبتنی بر درایور با میکروکنترلر ESP32

هدف پروژه: در این پروژه هنرجو یاد می‌گیرد چگونه یک موتور DC را به میکروکنترلر ESP32 متصل نماید و به صورت ساده کنترل کند.

مواد، ابزار و تجهیزات: لوازم التحریر، دستورهای زبان C، برگه اطلاعات برد توسعه ESP32، رایانه دارای Arduino IDE، شبیه‌ساز Wokwi، موتور DC، منبع تغذیه، بردبرد، سیم جامپر.

مراحل اجرای کار:

۱ مطابق با مدار شکل ۵۰، از درایور L۲۹۸ و منبع تغذیه برای راه‌اندازی موتور استفاده شده است. دقت شود که زمین منبع تغذیه و زمین میکروکنترلر باید به هم متصل شود.



شکل ۵۰- کنترل شدت نور LED به صورت آنالوگ توسط ولوم

۲ نرم‌افزار Arduino IDE را فعال نمایید و با زبان برنامه‌نویسی C برنامه زیر را بنویسید.

```
// Motor A
int motor1Pin1 = 27; //
int motor1Pin2 = 26; //
int enable1Pin = 25; //
void setup () {
    // sets the pins as outputs:
    pinMode(motor1Pin1, OUTPUT); //
    pinMode(motor1Pin2, OUTPUT); //
    pinMode(enable1Pin, OUTPUT); //
    Serial.begin(115200);
}
void loop() {
    // Move the DC motor forward at maximum speed
    Serial.println("Moving Forward"); //
    digitalWrite(enable1Pin, HIGH); //
    digitalWrite(motor1Pin1, LOW); //
    digitalWrite(motor1Pin2, HIGH); //
    delay(5000);
    // Stop the DC motor
    Serial.println("Motor stopped"); //
    digitalWrite(motor1Pin1, LOW); //
    digitalWrite(motor1Pin2, LOW); //
    delay(2000);
    // Move DC motor backwards at maximum speed
    Serial.println("Moving Backwards"); //
    digitalWrite(motor1Pin1, HIGH); //
    digitalWrite(motor1Pin2, LOW); //
    delay(5000);
    // Stop the DC motor
    Serial.println("Motor stopped"); //
    digitalWrite(motor1Pin1, LOW); //
    digitalWrite(motor1Pin2, LOW); //
    delay(2000);
}
```

۳ با توجه به دستورات آموزش داده شده، برنامه را تحلیل نمایید و در برنامه نوشته شده، مقابل خط‌های مشخص شده، توضیحات مناسب بنویسید.

در این پروژه، با استفاده از تنظیم پایه‌های خروجی ESP که به پایه‌های کنترلی L۲۹۸ متصل می‌باشد موتور به صورت «راست‌گرد - متوقف - چپ‌گرد - متوقف» عمل می‌کند.

۴ در صفحه وب https://docs.wokwi.com/Guides/ESP32_Simulator وارد قسمت Arduino Core قالب خام شبیه‌ساز (Arduino_ESP32 Project Template) را انتخاب نمایید. سپس مدار را در شبیه‌ساز بسته و اجرا نمایید.

۵ میکروکنترلر را به رایانه متصل نموده و پس از تنظیم نوع برد و پورت، میکروکنترلر را پروگرام و پروژه را اجرا نمایید.



پروژه زیر را با راهنمایی هنرآموز به صورت نرم افزاری (wokwi) و عملی اجرا نمایید.
برنامه‌ای بنویسید که در صورتی که یک میکروسوئیچ فشرده شود موتور به صورت راست گرد عمل کند و در غیر این صورت متوقف باشد.
برنامه‌ای بنویسید که از طریق دو میکروسوئیچ، بتوان موتور DC را کنترل کرد. اگر کلید اول فقط فشرده شود موتور راست گرد و اگر کلید دوم فشرده شود موتور چپ گرد عمل نماید و در غیر این دو حالت، موتور متوقف باشد.

خودآزمایی

- ۱ برای اندازه‌گیری کمیت‌هایی مانند دما و شدت نور که به صورت پیوسته تغییر می‌کنند، استفاده از کدام نوع ورودی ضروری است؟
(الف) ورودی دیجیتال (ب) ورودی آنالوگ (پ) ورودی وقفه (د) ورودی سریال
- ۲ وظیفه اصلی مبدل آنالوگ به دیجیتال (ADC) در ESP32 چیست؟
(الف) تبدیل سیگنال‌های دیجیتال به ولتاژ متغیر
(ب) تبدیل مقادیر پیوسته (آنالوگ) به داده‌های قابل فهم برای میکروکنترلر
(ج) تقویت جریان برای راه‌اندازی موتورهای DC
(د) حذف نویز از سنسورهای دیجیتال
- ۳ کدام یک از تکنیک‌های زیر برای تغییر پیوسته شدت نور LED یا سرعت موتور به کار می‌رود؟
(الف) GPIO (ب) ADC (پ) PWM (ت) Serial Monitor
- ۴ کدام قطعه برای کنترل موتور DC توسط میکروکنترلر استفاده می‌شود؟
(الف) درایور موتور (ب) مبدل ADC (پ) سنسور LDR (ت) پتانسیومتر (ولوم)
- ۵ ورودی‌ها و خروجی‌های دیجیتال به دلیل داشتن تنها دو حالت صفر و یک، برای اندازه‌گیری دقیق تمام کمیت‌های دنیای واقعی کافی هستند.
☐ درست ☐ نادرست
- ۶ کنترل سرعت موتورهای DC به صورت نرم و پیوسته، تنها با استفاده از ورودی‌های دیجیتال امکان‌پذیر است.
☐ درست ☐ نادرست
- ۷ هدف اصلی استفاده از مبدل ADC در ESP32، خواندن سنسورهایی مانند LDR و LM35 است که خروجی پیوسته دارند.
☐ درست ☐ نادرست
- ۸ دو مورد از سنسورهایی که مقادیر آنالوگ تولید می‌کنند، را نام ببرید.
- ۹ تفاوت بنیادین سیگنال دیجیتال با سیگنال آنالوگ چیست؟
- ۱۰ عبارت PWM مخفف چیست و چه کاربردی در کنترل تجهیزات دارد؟

ارزشیابی شایستگی ورودی و خروجی آنالوگ و کنترل موتور ESP32

نام و نام خانوادگی		شماره ملی	تاریخ ارزشیابی:	نوبت:
کد حرفه	۸۲۱۲۱۲	حرفه:	مونتازکاران تجهیزات و وسایل برقی و الکترونیکی	کد پیمانه/پودمان
کد وظیفه شغل	۸۲۱۲۱۲۰۳	وظیفه شغل:	ساخت مدارهای دیجیتال و میکروکنترلر	سطح صلاحیت
کد کار	۸۲۱۲۱۲۰۳۵۲	کار:	ساخت مدارهای کاربردی با (ESP32)	سطح شایستگی
				مهارت

ردیف	مراحل کار	شرایط انجام کار (ابزار، مواد، تجهیزات، زمان، مکان و ...)	ابزارهای ارزشیابی	استاندارد (شاخص ها/داوری/نمره دهی)	نتایج ممکن	نمره کسب شده
۱	شناسایی پایه‌های ADC و DAC در ESP32 و تنظیم صحیح آنها در کد برای ورودی یا خروجی بودن	رایانه دارای Arduino IDE، برد ESP32، کابل USB، برد بُرد، سنسور آنالوگ یا پتانسیومتر، سیم اتصال، میز کار کارگاه الکترونیک، زمان ۲۰- ۱۵ دقیقه	مشاهده عملکرد، بررسی نحوه سیم‌کشی مدار و کد برنامه در IDE، چک‌لیست شناسایی و تنظیم پایه‌های ADC و DAC	شناسایی و پیکربندی صحیح پایه‌های ADC و DAC در کد	۳	
				انجام کار با خطای جزئی	۲	
				ناتوانی در تشخیص و تنظیم پایه‌ها	۱	
۲	دریافت و پردازش صحیح سیگنال‌های حسگرها (مثل ولتاژ یا دما)	رایانه دارای Arduino IDE، برد ESP32، کابل USB، برد برد، سنسور آنالوگ یا پتانسیومتر، سیم اتصال و زمان ۲۰- ۱۵ دقیقه در کارگاه الکترونیک	مشاهده عملکرد، بررسی سیم‌کشی، اتصالات مدار و کد برنامه در IDE، چک‌لیست دریافت و پردازش داده‌ها	دریافت و پردازش صحیح داده‌ها	۳	
				انجام کار با خطای جزئی	۲	
				ناتوانی در خواندن یا پردازش	۱	
۳	تولید سیگنال PWM یا ولتاژ آنالوگ برای کنترل دقیق وسایلی مانند سرعت موتور یا روشنایی	رایانه دارای Arduino IDE، برد ESP32، منبع تغذیه، LED یا موتور DC، مقاومت و سیم رابط و زمان ۳۰- ۲۰ دقیقه در کارگاه الکترونیک	مشاهده عملکرد مدار، بررسی کد برنامه، اندازه‌گیری سیگنال خروجی با مولتی‌متر یا اسیلوسکوپ، چک‌لیست ارزیابی عملکرد	راه‌اندازی و اجرای صحیح مدار و کنترل دقیق خروجی	۳	
				اجرا با خطای جزئی یا کنترل محدود	۲	
				عدم راه‌اندازی صحیح یا عملکرد نادرست سیستم	۱	
۴	ترکیب ورودی و خروجی‌ها در یک پروژه عملی با کدنویسی، مدیریت کنترل خطا و درک نقش آن در هوشمندسازی سامانه، مستندسازی و گزارش‌نویسی پروژه	رایانه دارای Arduino IDE، برد ESP32، منبع تغذیه، LED یا موتور DC، مقاومت و سیم‌های رابط و زمان ۶۰- ۴۵ دقیقه در کارگاه الکترونیک	مشاهده عملکرد، بررسی کد برنامه و ساختار آن و گزارش مکتوب، چک‌لیست ارزیابی پروژه	پیاده‌سازی کامل سیستم با عملکرد صحیح	۳	
				پیاده‌سازی با خطای جزئی	۲	
				عدم تکمیل پروژه یا عملکرد نامناسب سیستم	۱	
	شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش:	کارگاه الکترونیک یا سایت رایانه، تجهیزات کارگاهی و حفاظت فردی و دستورالعمل‌های ایمنی و بهداشت حرفه‌ای	مشاهده رفتار هنرجو در حین انجام فعالیت، میزان رعایت اصول ایمنی و نظم، فعالیت‌های گروهی، مدیریت زمان و مسئولیت‌پذیری	رعایت نسبی اصول ایمنی، بهداشت و ملاحظات زیست‌محیطی، با خطاهای جزئی در حین انجام کار.	۲	
				عدم رعایت اصول ایمنی، بهداشت و ملاحظات زیست‌محیطی، یا بروز خطاهای جدی در نگرش و رفتار کاری.	۱	
						☐ بلی
						☐ خیر
ارزشیابی کار (شایستگی انجام کار)						

توضیحات:

۱- معیار شایستگی انجام کار:

- کسب حداقل نمره ۲ مستلزم راه‌اندازی صحیح ورودی‌ها، پردازش منطقی داده‌ها، کنترل قابل قبول خروجی‌ها و پیاده‌سازی قابل قبول پروژه با مستندسازی و ارائه پایه‌ای است (مراحل بحرانی) ..
- کسب حداقل نمره ۲ از بخش شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش
- کسب حداقل میانگین ۲ از مراحل کار

۲- تعریف سطوح شایستگی: صرف نظر از اینکه یک تکلیف کاری در چه سطحی از صلاحیت حرفه‌ای انجام می‌شود، در هر محیط کاری ممکن است انجام هر کار با کیفیت مشخصی مورد انتظار باشد. سطح شایستگی انجام کار، معیار اساسی ارزشیابی است.

مهارت (سطح سه): ماهر و قادر به آموزش و هدایت دیگران، توانایی برنامه‌ریزی و تحلیل، پاسخ‌گویی در برابر کارهای خود، سروکار داشتن با سطح وسیعی از کارها و فعالیت‌ها تسلط (سطح چهار): خبرگی در انجام کار و آموزش دیگران، ایجاد نوآوری، سازگاری، عیب‌یابی، هدایت و راهنمایی دیگران.

۱ مونتاز و دمونتاز اس.ام. دی و مستندسازی، کد ۲۱۱۲۷۶، مؤلفان: محمود شبانی، محمد خیجی، مجید مکرم، زانکو طهماسبی، علیرضا گنجه‌خور، سارا ساوئی، انتشارات شرکت چاپ و نشر کتاب‌های درسی ایران ۱۴۰۴.

۲ دانش فنی تخصصی، کد ۲۱۲۲۷۵، مؤلفان: سیدمحمود صموتی، شهرام نصیری سوادکوهی، رسول ملک‌محمد، محمود شبانی، رضا فتاحی، احمد توکلی، انتشارات شرکت چاپ و نشر کتاب‌های درسی ایران ۱۴۰۴.

۳ مبانی دیجیتال، کد ۴۸۹/۸، یدالله رضازاده، غلامحسین نصری، فتح‌الله نظریان، مهین ظریفیان جولایی، رسول ملک‌محمد، محمود شبانی، انتشارات شرکت چاپ و نشر کتاب‌های درسی ایران، ۱۳۹۴.

۴ آزمایشگاه مجازی ۲، کد ۴۶۶/۶، مؤلفان: مهین ظریفیان جولایی، سیدمحمود صموتی، سیدعلی صموتی، محمود شبانی، شرکت چاپ و نشر کتاب‌های درسی ایران، ۱۳۹۴.

۵ سایت‌های اینترنتی، برگه‌های اطلاعات Data sheet، کتاب اطلاعات Data book و دستینه Hand book

۶ سایت آموزش‌های تخصصی و پروژه‌محور ESP32

randomherd tutorials.com

۷ Patterson, D. A., & Hennessy, J. L. Computer Organization and Design: The Hardware/Software Interface (6th ed.). Morgan Kaufmann, 2021.

۸ Martin, R.C. Clean Code: A Handbook of Agile Software Craftsmanship. Prentice Hall, 2008.

۹ Monk, S. Programming Arduino: Getting Started with Sketches (2nd ed.). McGraw-Hill Education, 2016.

۱۰ Espressif Systems. ESP-IDF Programming Guide.

<https://docs.espressif.com>

۱۱ https://documentation.espressif.com/esp32_technical_reference_manual_en

۱۲ Wokwi Documentation.

<https://docs.wokwi.com>

۱۳- تجربیات، دستاوردها و پروژه‌های عملی طراحی و پیاده‌سازی‌شده در حوزه برنامه‌نویسی میکروکنترلرها، سامانه‌های نهفته (Embedded Systems) و اینترنت اشیا (IoT)

