



فصل ۴

میکروکنترلرها Microcontrollers



میکروکنترلر یک مدار مجتمع یا چیپ الکترونیکی است که دارای پردازنده حافظه و تعدادی ورودی و خروجی قابل برنامه‌ریزی است. میکروکنترلرها در واقع یک میکروکامپیوتر هستند که از نظر عملکرد تقریباً مشابه یک کامپیوتر کامل می‌باشند. میکروکنترلرها دارای ورودی، پردازش و خروجی هستند که برای مصارف خاصی برنامه‌ریزی می‌شوند. میکروکنترلرها در طراحی مدارهای کنترلی استفاده می‌شوند. طراحی مدارهای کنترلی که قبلاً با صدها مدار مجتمع آنالوگ و دیجیتال به اجرا در می‌آمد امروزه با یک میکروکنترلر ساده قابل انجام است.

واحد یادگیری ۶

میکروکنترلر و چگونگی برنامه‌ریزی آن

آیا تا به حال فکر کرده‌اید

- یک میکروکنترلر از چه اجزایی تشکیل شده است؟
 - ابعاد یک میکروکنترلر چقدر است؟
 - چند نوع میکروکنترلر وجود دارد؟
 - استفاده از میکروکنترلرها چه مزیت‌هایی دارد؟
 - چه زبان‌های برنامه‌نویسی برای میکروکنترلرها وجود دارند؟
 - چگونه می‌توان برنامه نوشته شده را به زبان قابل فهم میکروکنترلر تبدیل کرد؟
 - چگونه می‌توان برنامه نوشته شده را داخل حافظه میکروکنترلر بارگذاری کرد؟
- کلمه میکروکنترلر از دو کلمه میکرو و کنترل تشکیل شده است: منظور از میکرو می‌تواند سرعت عمل میکروکنترلر باشد، زیرا میکروکنترلر می‌تواند دستورهایی را که به آن می‌دهیم با سرعت بالا و در حد زمان یک میکرو ثانیه انجام بدهد. معنی دوم میکرو شاید کوچکی قطعه باشد، زیرا این قطعه با ابعادی تا یک میلیونوم متر کوچک شده است. شاید باورکردنی نیست که در یک تراشه ممکن است بیش از یک میلیون ترانزیستور به کار رفته باشد. معنی فارسی میکروکنترلر را ریزکنترلر یا ریزکنترل کننده نامیده‌اند.
- میکروکنترلرها قابل برنامه‌ریزی هستند، به طوری که طراح می‌تواند با توجه به نیاز کاربر آن را برنامه‌ریزی کند. به عنوان مثال با اتصال کی‌بورد و آل‌سی‌دی به میکروکنترلر برنامه‌ریزی شده، یک ماشین حساب یا قفل رمز دیجیتالی بسازد. میکروکنترلرها به طور گسترده در تولید سیستم‌های تک منظوره مانند سیستم‌های حفاظتی، دزدگیر، سیستم‌های کنترل صنعتی، دستگاه دورنگار (فاکس)، فرکانس متر، کنترل موتور ربات و ولت متر DC به کار می‌روند.

استاندارد عملکرد

میکروکنترلر و چگونگی برنامه‌ریزی آن با رعایت استاندارد تعریف شده

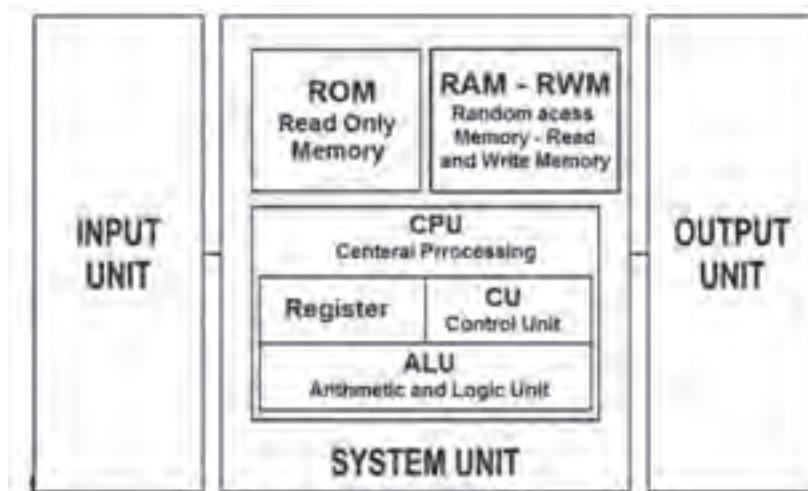
۶-۱ مواد، ابزار و تجهیزات مورد نیاز

میکروکنترلر (AVR ATmega) - رایانه - نرم افزار پروتئوس Proteus, Bascom, Code Vision AVR - Ver. ۱۲-۳ - فرهنگ لغت انگلیسی به فارسی - ترانزیستور BC ۱۴۰ - دیود ۱N۴۰۰۱ - مقاومت ۱ K Ω ، ۱/۴ وات - لامپ ۱۲ ولت - رله ۱۲ ولتی یک کنتاکت - ترانسفورماتور V ۶/۲۰-۲۲۰ - موتور ۱۲ ولتی کوچک و کم وات

۶-۲ معرفی میکروکنترلرها و شرح عملکرد آنها

سیستم کامپیوتری

در سال های گذشته با ساختار کلی رایانه (کامپیوتر) آشنا شده اید، اجزای تشکیل دهنده یک سامانه کامپیوتری به صورت بلوک دیاگرام شکل ۶-۱ نشان داده شده است.



شکل ۶-۱ ساختار کلی یک سامانه رایانه (کامپیوتر)

واژه های داخل تصویر ۶-۱ را به زبان فارسی ترجمه کنید.

ترجمه کنید



در داخل دستگاه هایی مانند کنترل از راه دور، دستگاه صوتی و تصویری، ربات، کنترل دمای یخچال و کنترل ماشین لباسشویی از یک رایانه استفاده می شود. در مورد آنها جست و جو کنید و نتیجه را از طریق بارش فکری در کارگاه به بحث بگذارید.

بارش فکری



با مراجعه به رسانه های مختلف کلمات Micro controller block Diagram را جست و جو کنید و دو نمونه دیگر از بلوک دیاگرام سامانه کامپیوتری را بیابید و آنها را با هم مقایسه کنید، نتیجه را به صورت یک گزارش به کارگاه ارائه دهید.

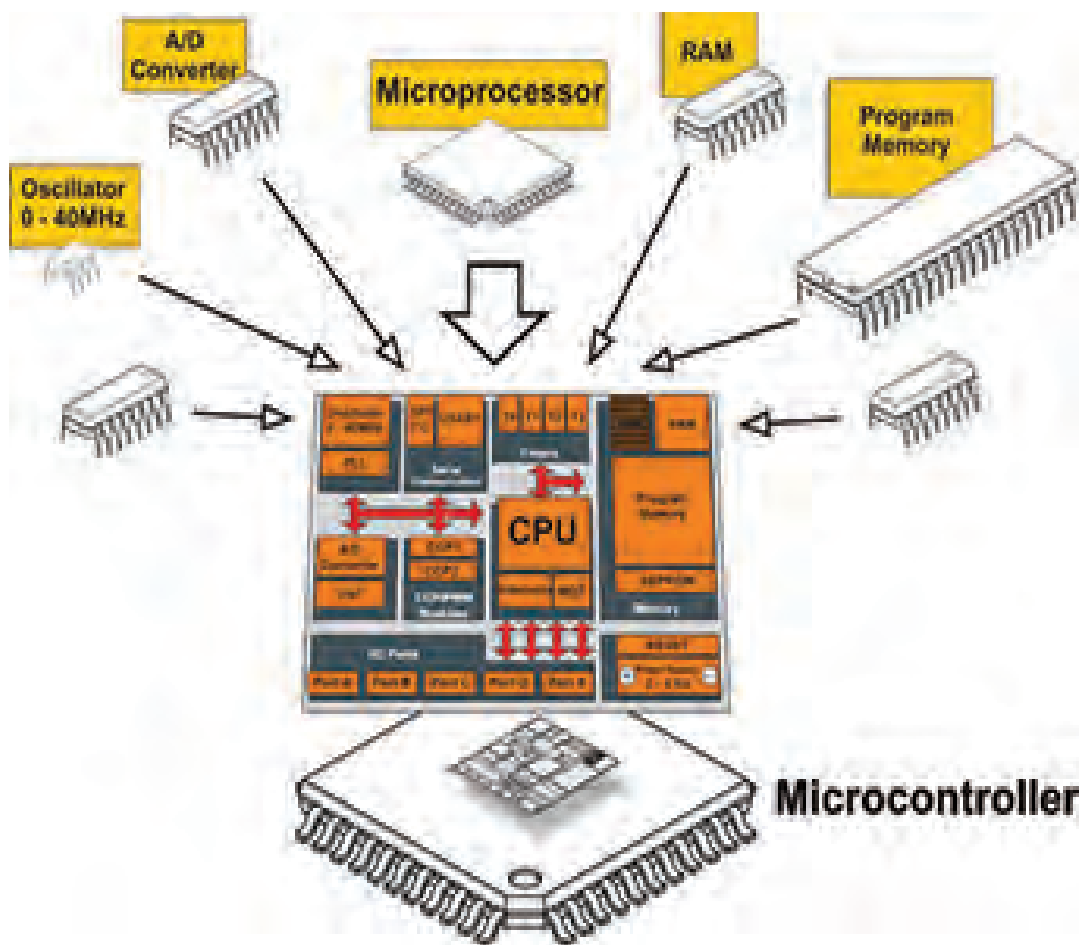
فعالیت



■ معرفی میکروکنترلر و میکرو کامپیوترها

با وجود اینکه میکروکنترلرها (ریزکنترل گرها) و میکروپروسسورها (ریزپردازنده‌ها) را از نظر فنی و ساختاری در یک شاخه قرار می‌دهند ولی آنها در اجزاء و عملکرد با هم تفاوت‌های عمده دارند. در ریزپردازنده‌ها فقط واحد پردازش به صورت مجزا به صورت یک آی‌سی IC (مدار مجتمع) وجود دارد که مدارهایی مانند RAM، ROM، واحد ورودی/خروجی، تایمرها و ثبات‌ها در آی‌سی‌های مجزا به ریزپردازنده متصل می‌شوند و تشکیل یک رایانه را می‌دهند.

در ریزکنترل گر تمام واحدهای یک رایانه، از جمله حافظه‌های RAM، ROM، تایمرها و ثبات‌ها همگی به‌صورت فشرده و یکپارچه در درون یک آی‌سی به نام میکروکنترلر قرار می‌گیرند شکل ۲-۶. امروزه استفاده از میکروکنترلرها در دستگاه‌های تجاری برای کنترل عملکرد دستگاه‌های الکترونیکی خانگی مانند تلویزیون، اجاق میکروویو (Microwave oven)، دستگاه کنترل از راه دور و ماشین لباسشویی بسیار متداول و مقرون به‌صرفه است. به همین دلیل میکروکنترلر را میکرو کامپیوتر هم می‌نامند.



شکل ۲-۶ ساختار کلی میکروکنترلر

ارزشیابی نظری تشخیصی

نکته



این ارزشیابی توسط هنرجو در ساعات غیردرسی اجرا شود و نتیجه ارزشیابی به مربی مربوطه ارائه گردد. محتوای این ارزشیابی نظری مجدداً در کلاس درس توسط هنرآموز مربوطه به اجرا در می‌آید و بر اساس سطح کلاس روند آموزش ادامه خواهد یافت.

۱ آیا هزینه طراحی مدار کنترل یک دستگاه خانگی مانند ماشین لباسشویی به وسیله رایانه یا طراحی آن توسط میکروکنترلر برابری می‌کند؟ توضیح دهید.

۲ آیا یک رایانه از نظر حجم داده و سرعت پردازش (محاسبات) با یک میکروکنترلر برابری می‌کند، کدام بیشتر است؟

۳ در شکل ۶-۲ ساختار یک میکروکنترلر قسمت A/D Converter به کدام بخش اشاره دارد؟

الف) بخش محاسبه و منطق (ب) مودم و ارتباط شبکه

ج) ورودی حافظه جانبی (د) مبدل آنالوگ به دیجیتال

۴ در شکل ۶-۲ میکروکنترلر Program memory به کدام حافظه اشاره دارد؟

الف) هارد دیسک (ب) حافظه‌های فقط خواندنی

ج) حافظه جانبی (Micro SD) (د) حافظه دینامیک و استاتیک

۵ به کدام علت در یک ماشین لباسشویی نمی‌توان از یک رایانه کامل معمولی استفاده کرد؟

الف) وزن و حجم زیاد (ب) هزینه زیاد

ج) عدم نیاز به حجم زیاد حافظه و پردازش (د) هر سه مورد

۶ برای یک دستگاه کنترل از راه دور با ۱۲ عدد کلید، در زمانی که یک کلید زده می‌شود، حداقل کد

ارسالی چند بیتی (چند رقمی) است؟

الف) ۱۲ (ب) ۴ (ج) ۶ (د) ۸

۷ امروزه برای کنترل دستگاه‌ها (خانگی و تجاری) از مدارهای دیجیتالی پایه مانند گیت‌ها، دیکودرها،

فلیپ‌فلاپ‌ها استفاده نمی‌شود، چرا؟

الف) نیاز به مدارچاپی بزرگ (ب) قیمت بالا

ج) طراحی سخت و تغییرناپذیر (د) هر سه مورد

فیلم کاربردی از میکروکنترلرها را ببینید.

فیلم



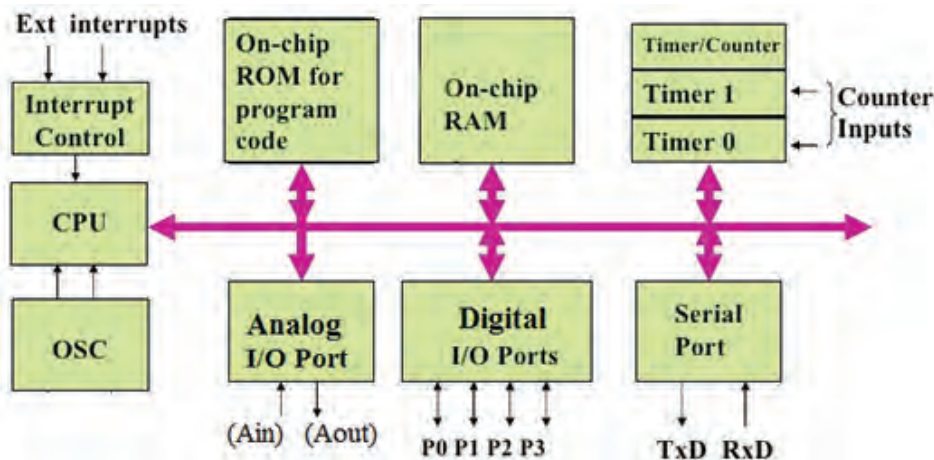
درباره علت عدم استفاده از رایانه معمولی به جای میکروکنترلر در دستگاه‌های الکترونیکی بررسی کنید و نتیجه را از طریق بارش فکری به بحث بگذارید.

بارش فکری



۳-۶ معماری میکروکنترلر

چگونگی قرار گرفتن بخش‌های مختلف میکروکنترلر مانند مبدل A/D، Timer/counter، RAM، ROM، CPU و در کنار هم را معماری میکروکنترلر می‌گویند. معماری میکروکنترلر زمانی کامل است که اجزای آن روند مناسبی را برای اجرای دستورهای موجود در آن دنبال کنند. شکل ۳-۶ بخش‌های اصلی یک میکروکنترلر را به صورت بلوک دیاگرام نشان می‌دهد. هماهنگی و نظم در چیدمان اجزای میکروکنترلر، نقش مؤثری در سرعت اجرای دستورهای آن دارد. مثلاً سرعت اجرای دستورهای میکروکنترلرهای ۸۰۵۱ به میزان ۱۲ برابر از میکروکنترلرهای AVR کمتر است. علت این تفاوت در معماری این دو نوع میکروکنترلر است.



شکل ۳-۶ معماری بخش‌های اصلی یک میکروکنترلر

با مراجعه به رسانه‌های مختلف کلمات computer architector block diagram را جست‌وجو کنید و نمونه‌های دیگری از نقشه بلوکی معماری کامپیوتر را بیابید و نتیجه را در قالب گزارش پژوهشی به کارگاه ارائه کنید.

فعالیت



■ اجزای اصلی میکروکنترلر

میکروکنترلرهای امروزی از نظر معماری دارای بخش‌های مشترکی به شرح زیر هستند:

- ✓ **حافظه موقتی RAM:** در این حافظه برنامه و بخشی از داده (Data) موقت به اجرا درمی‌آید و به صورت موقت تا زمانی که میکروکنترلر به جریان الکتریکی متصل است، ذخیره می‌شود.
- ✓ **حافظه خواندنی ROM:** این حافظه جهت قرار گرفتن برنامه مورد نیاز و داده‌هایی که ثابت هستند استفاده می‌شود. این داده‌ها شامل جدولی از داده‌های ثابت است که برای نمایش کاراکتر و شکل‌های خاص در نمایشگری که سون سگمنت (V Seg) به میکروکنترلر اتصال داده شده است به کار می‌رود. امروزه در میکروکنترلرها، از خانواده فلش رام (Cooldisk Flash RAM) استفاده می‌شود. اگر در فرایند اجرای برنامه نیاز به ذخیره اطلاعات پایدار بدون نیاز به تغذیه باشد از حافظه‌های EEPROM نیز استفاده می‌شود. در این حالت با قطع شدن تغذیه، اطلاعات ذخیره شده در میکروکنترلر از بین نمی‌رود.

✓ **واحد پردازش مرکزی – CPU:** به معنی پردازشگر است و واحد اصلی و مرکزی (مغز) میکروکنترلر به حساب می‌آید. پردازشگر از بخش‌های محاسبه، منطق، رجیسترهای اصلی و واحد دریافت و ترجمه کدهای برنامه تشکیل شده است. هر قدر هماهنگی این بخش‌ها با هم بیشتر باشد عمل پردازش سریع‌تر صورت می‌گیرد، این نوع میکروکنترلرها را میکروکنترلر سرعت بالا می‌نامند.

✓ **ورودی و خروجی دیجیتالی – DIGITAL I/O:** این بخش شامل مجموعه پایه‌های ورودی و خروجی میکروکنترلر است، عملکرد این پایه‌ها از طریق دستورهای تدوین شده در برنامه‌نویسی قابل تعریف است. به این ترتیب می‌توانیم به ورودی یا خروجی داده‌های (Data) مختلف را اختصاص دهیم و حالت‌های مختلف را به وجود آوریم.

✓ **ساعت – Clock:** در این بخش یک موج مربعی به عنوان پالس ساعت تولید می‌شود. این پالس مورد نیاز میکروکنترلر است، که برای مثال می‌تواند به دو صورت زیر تولید شود:

- حالت RC داخلی: پالس ساعت در داخل تراشه (آی سی IC) میکروکنترلر تولید می‌شود.
- حالت کریستال خارجی: با نصب یک کریستال پیزو الکتریک به پایه‌های میکروکنترلر پالس ساعت تولید می‌شود. در اکثر کاربردها از حالت RC داخلی استفاده می‌شود، زیرا دیگر نیاز به قطعه خارجی برای اتصال به میکروکنترلر ندارد.

در اصطلاح عمومی «میکروکنترلرها» را «میکرو» نیز می‌نامند. بنابراین در مراحل بعد ممکن است ما نیز از اصطلاح میکرو استفاده کنیم.

نکته



در خارج از ساعات درسی و با نظارت اولیای خود، با مراجعه به سایت‌های الکترونیک درباره کاربردهای Timer و Counter و A/D (آنالوگ به دیجیتال) جست‌وجو کنید و در کلاس ارائه کنید.

جست‌وجو کنید



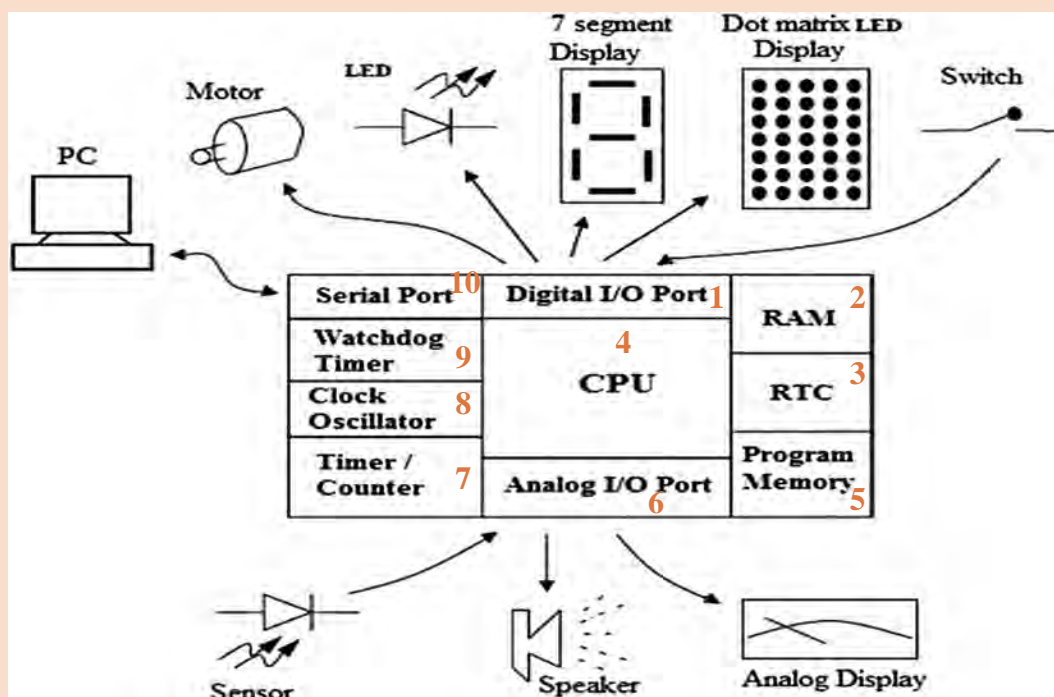
ترجمه اطلاعات فنی و بخش‌های مختلف میکروکنترلر

هدف: کسب مهارت در شناخت و تشخیص اجزاء مختلف یک میکروکنترلر از یکدیگر
مواد، ابزار و تجهیزات: لوازم التحریر- فرهنگ لغت انگلیسی به فارسی
مراحل اجرای کار:

1 با بررسی شکل ۴-۶، بلوک دیاگرام قسمت‌های اصلی و مهم آی سی میکروکنترلر را ترجمه کنید و جدول ۱-۶ را کامل کنید.

کار عملی ۱





شکل ۶-۴ ساختار کلی یک میکروکنترلر

جدول ۶-۱

ترجمه	شماره بلوک	عنوان بلوک
	۱	Digital I/O Port
		Analog I/O Port
		RAM (Random Access Memory)
		RTC(Real Time Clock)
		Program Memory
		Serial Port
		Watchdog Timer
		Clock Oscillator
		Timer/Counter

۲ با توجه به شکل ۶-۴ قطعات جانبی که به درگاه‌های ورودی خروجی (I/O Ports) میکروکنترلر اتصال داده می‌شوند را تعیین کنید و در جدول ۶-۲ بنویسید.

جدول ۶-۲

ردیف	نام دستگاه یا قطعه	ورودی برای میکروکنترلر	خروجی برای میکروکنترلر
۱	Personal Computer(PC)	به پورت سریال میکروکنترلر اتصال داده می شود	————
۲			
۳			
۴			
۵			
۶			
۷			
۸			
۹			

۳ با مراجعه به فرهنگ لغت یا سایر رسانه‌ها، اصطلاحات داده شده در جدول ۶-۳ را ترجمه کنید و نماد اختصاری آن را بنویسید. سپس این نماد اختصاری را با سایر نمادهای اختصاری که تا کنون آموخته‌اید مقایسه کنید و در صورت مطابقت در ستون مربوطه علامت (✓) و در صورت عدم تطابق علامت (-) را وارد کنید.

جدول ۶-۳

تطابق	نماد اختصاری	واژه به زبان فارسی	واژه به زبان انگلیسی	ردیف
			Random Access Memory	۱
			Read Only Memory	۲
			Erasable Read Only Memory	۳
			Erasable Programmable Read Only Memory	۴
			Electrically Erasable Programmable Read Only Memory	۵
			Analog to Digital	۶
			Central Processing Unit	۷
			Control Unit	۸
			Arithmetic Processing Unit	۹



- ۱ مجموعه‌ای از قطعات الکترونیک در یک بسته‌بندی را می‌گویند.
- ۲ مجموعه‌ای از مدارهای مجتمع را که در یک بسته قرار دارند و مانند یک رایانه کوچک عمل می‌کنند را می‌گویند.
- ۳ در شکل ۶-۴ Analog I/O Port شامل چه بخش‌هایی می‌شود؟
- ۴ به میکروکنترلر، ریزکنترل گر نیز می‌گویند. درست □ نادرست □
- ۵ در شکل ۶-۴ Timer برای چه منظوری استفاده می‌شود؟

۶-۴ انواع میکروکنترلر

پیشرفت فناوری در صنعت الکترونیک، و ارتباط آن با علم دیجیتال، سبب رشد چشمگیری در شکل‌گیری ساختار انواع خانواده میکروکنترلرها شده است. کارخانه‌های مختلف با توجه به نیاز، انواع میکروکنترلرها را تولید می‌کنند. محصول تولیدی آنها در پروژه‌های صنعتی، در بین متخصصین، دانشجویان و علاقه‌مندان رواج بیشتری دارد. متداول‌ترین میکروکنترلرها خانواده‌های ۸۰۵۱، AVR، PIC و ARM هستند.

فیلم انواع مختلف میکروکنترلرها و کاربرد آنها را ببینید.

فیلم



■ خانواده ۸۰۵۱:

این خانواده از قدیمی‌ترین میکروکنترلرها به شمار می‌آیند، که امکانات محدودی دارند. این نوع میکروکنترلرها قیمت بسیار پایینی دارند. شکل ۶-۵ قسمتی از برگه اطلاعات آی سی ۸۰۵۱ را نشان می‌دهد.



شکل ۶-۵ برگه اطلاعات میکروکنترلر ۸۰۵۱

موارد شماره‌گذاری شده در شکل ۶-۵ را ترجمه کنید و نتیجه را در کلاس به بحث بگذارید.

ترجمه کنید





یک نمونه آی سی ۸۰۵۱ را در اختیار بگیرید و مشخصات ظاهری آن را با انواع آی سی هایی که تا کنون دیده‌اید مقایسه کنید.

■ خانواده AVR

این خانواده از رایج‌ترین و پرکاربردترین میکروکنترلرها است که در مدل‌های مختلف با امکانات متنوع ساخته می‌شوند و در بازار با قیمت مناسب در دسترس است. شکل ۶-۶ قسمتی از برگه اطلاعات یک نمونه میکروکنترلر AVR را نشان می‌دهد.

<p>Features</p> <ul style="list-style-type: none"> • High-performance, Low-power AVR[®] 8-bit Microcontroller • Advanced RISC Architecture <ul style="list-style-type: none"> – 130 Powerful Instructions – Most Single-clock Cycle Execution 1 – 32 x 8 General Purpose Working Registers 2 – Fully Static Operation – Up to 16 MIPS Throughput at 16 MHz 3 – On-chip 2-cycle Multiplier • Nonvolatile Program and Data Memories <ul style="list-style-type: none"> – 8K Bytes of In-System Self-Programmable Flash 4 <ul style="list-style-type: none"> Endurance: 10,000 Write/Erase Cycles – Optional Boot Code Section with Independent Lock Bits – In-System Programming by On-chip Boot Program – True Read-While-Write Operation – 512 Bytes EEPROM 5 <ul style="list-style-type: none"> Endurance: 100,000 Write/Erase Cycles – 1K Byte Internal SRAM 6 – Programming Lock for Software Security • Peripheral Features <ul style="list-style-type: none"> – Two 8-bit Timer/Counters with Separate Prescaler, one Compare Mode – One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode – Real Time Counter with Separate Oscillator – Three PWM Channels – 8-channel ADC in TQFP and MLF package <ul style="list-style-type: none"> Eight Channels 10-bit Accuracy – 6-channel ADC in PDIP package 8 <ul style="list-style-type: none"> Eight Channels 10-bit Accuracy – Byte-oriented Two-wire Serial Interface – Programmable Serial USART – Master/Slave SPI Serial Interface – Programmable Watchdog Timer with Separate On-chip Oscillator – On-chip Analog Comparator 9 • Special Microcontroller Features <ul style="list-style-type: none"> – Power-on Reset and Programmable Brown-out Detection – Internal Calibrated RC Oscillator 10 – External and Internal Interrupt Sources – Five Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, and Standby • I/O and Packages <ul style="list-style-type: none"> – 23 Programmable I/O Lines 11 – 28-lead PDIP, 32-lead TQFP, and 32-pad MLF • Operating Voltages <ul style="list-style-type: none"> – 2.7 - 5.5V (ATmega8L) – 4.5 - 5.5V (ATmega8) 12 • Speed Grades <ul style="list-style-type: none"> – 0 - 8 MHz (ATmega8L) – 0 - 16 MHz (ATmega8) 13 • Power Consumption at 4 MHz, 3V, 25°C <ul style="list-style-type: none"> – Active: 3.6 mA – Idle Mode: 1.0 mA 14 – Power-down Mode: 0.5 µA 	 <p>8-bit AVR[®] with 8K Bytes In-System Programmable Flash</p> <hr/> <p>ATmega8 ATmega8L</p>
--	---

شکل ۶-۶ برگه اطلاعات میکروکنترلر خانواده AVR

ترجمه کنید



فعالیت خارج از ساعات درسی



موارد شماره گذاری شده در شکل ۶-۶ را ترجمه کرده و در کلاس درباره آنها بحث کنید.

سایر قسمت های مربوط به شکل ۶-۶ را در خارج از محیط آموزشی ترجمه کنید و نتیجه را به کارگاه ارائه دهید.

■ خانواده PIC

این خانواده نیز از رایج ترین و پر کاربرد ترین میکروکنترل ها است. میکروکنترلر PIC در مقایسه با میکروکنترل AVR دستورهای کمتری دارد، بنابراین حافظه کمتری را در اختیار می گیرد. این امر باعث افزایش سرعت پردازش PIC می شود. PIC مدل های مختلف و امکانات متنوع دارد و از قیمت مناسبی نیز برخوردار است. شکل ۶-۷ یک نمونه برگه اطلاعات میکروکنترلر PIC۱۶C۷۲ را نشان می دهد.

MICROCHIP **PIC16C72 SERIES**
8-Bit CMOS Microcontrollers with A/D Converter

Devices included:

- PIC16C72
- PIC16CR72

Microcontroller Core Features:

- High-performance RISC CPU 1
- Only 35 single word instructions to learn 2
- All single cycle instructions except for program branches which are two cycle 3
- Operating speed: DC - 20 MHz clock input 4
DC - 200 ns instruction cycle
- 2K x 14 words of Program Memory, 5
128 x 8 bytes of Data Memory (RAM) 6
- Interrupt capability
- Eight level deep hardware stack
- Direct, indirect, and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRIT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode 7
- Selectable oscillator options
- Low-power, high-speed CMOS technology
- Fully static design
- Wide operating voltage range:
 - 2.5V to 6.0V (PIC16C72) 8
 - 2.5V to 5.5V (PIC16CR72)
- High Sink/Source Current 25/25 mA
- Commercial, Industrial and Extended temperature ranges 9
- Low-power consumption:
 - < 2 mA @ 5V, 4 MHz
 - 15 μ A typical @ 3V, 32 kHz
 - < 1 μ A typical standby current 10

Pin Diagrams

SDIP, SOIC, SSOP, Windowed Side Braided Ceramic

PIC16C72
PIC16CR72

Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during sleep via external crystal/clock 11
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Capture, Compare, PWM (CCP) module
 - Capture is 16-bit, max. resolution is 12.5 ns
 - Compare is 16-bit, max. resolution is 200 ns
 - PWM max. resolution is 10-bit
- 8-bit 5-channel analog-to-digital converter 12
- Synchronous Serial Port (SSP) with SPI[™] and I²C[™]
- Brown-out detection circuitry for Brown-out Reset (BOR)

شکل ۶-۷ برگه اطلاعات میکروکنترلر خانواده PIC

ترجمه کنید



موارد شماره گذاری شده در شکل ۶-۷ را ترجمه کنید و نتیجه را در کارگاه به بحث بگذارید.

سایر قسمت‌های مربوط به شکل ۶-۷ را در خارج از محیط آموزشی ترجمه کنید و نتیجه را در قالب یک پژوهش به کلاس ارائه دهید.

فعالیت در
ساعات غیر
درسی

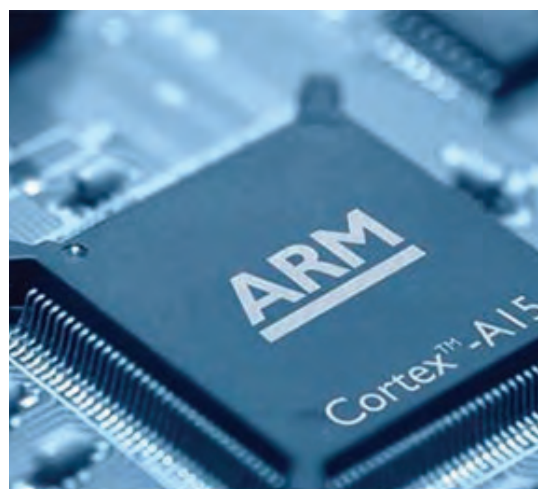


■ خانواده ARM

این خانواده فقط در بسته‌های SMD و در مدل‌های ۸، ۱۶، ۳۲ و ۶۴ بیتی ساخته می‌شوند. آی‌سی‌های ۳۲ بیتی این خانواده بیشترین کاربرد را دارند. امروزه میکروکنترلرهای ARM در بسیاری از وسایل مانند کامپیوترهای کوچک، تلفن همراه و تبلت استفاده می‌شوند. شاید بتوان گفت در آینده این میکروکنترلرها نقش پررنگی را در صنایع و جامعه خواهند داشت. قیمت این میکروکنترلرها با توجه به توانایی و امکاناتی که دارند بسیار پایین است. (شکل الف) ۶-۸ تصویر واقعی و (شکل ب) ۶-۸ ساختار داخلی میکروکنترلر ARM را نشان می‌دهد.



ب) ساختار داخلی میکروکنترلر ARM



الف) تصویر واقعی میکروکنترلر ARM

شکل ۶-۸ میکروکنترلر ARM

موارد شماره‌گذاری شده در شکل ۶-۸ را ترجمه کرده در کلاس درباره آنها بحث کنید.

ترجمه کنید



با استفاده از آی‌سی واقعی و روی بردهای اوراقی، چند نمونه آی‌سی میکروکنترلر را شناسایی کنید و نوع خانواده آن را تعیین نمایید.

فعالیت



با مراجعه به منابع مختلف از جمله سایت‌های اینترنتی، برگه‌های اطلاعات میکروکنترلرها را دانلود کنید و آنها را از نظر ساختار، کاربرد، امکانات داخلی و قیمت مقایسه کنید و نتایج را در کارگاه به بحث بگذارید.

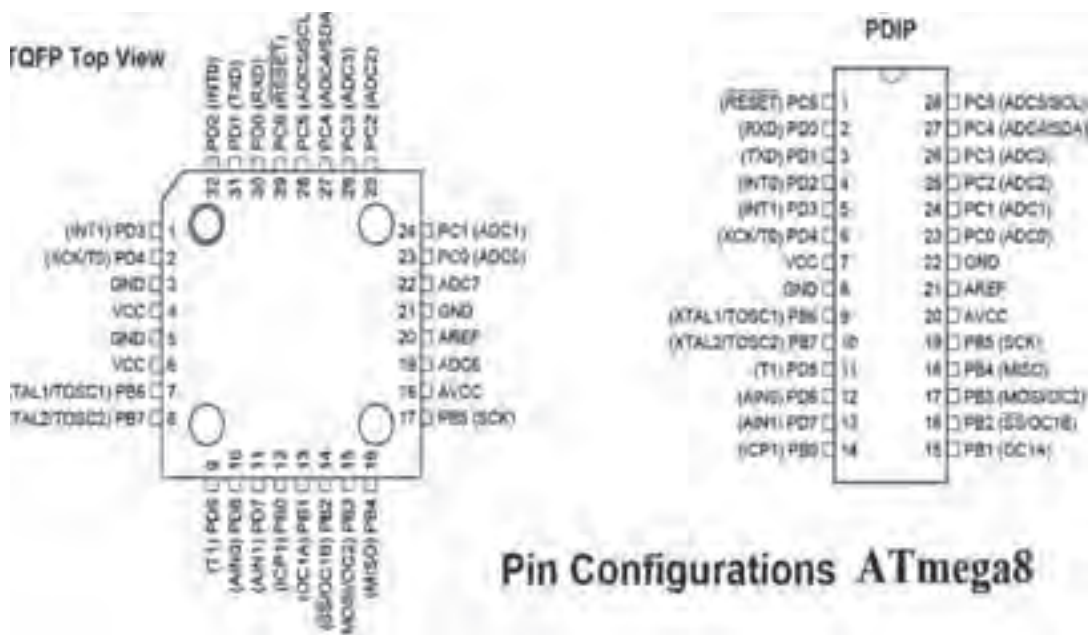
تحقیق کنید



۵-۶ انتخاب میکروکنترلر مناسب برای یک پروژه مدار الکترونیکی ساده

بخش‌های مختلف میکروکنترلر هر کدام برای کاربردی خاص در آن در نظر گرفته شده‌اند. یک طراح الکترونیک با توجه به نیازی که در پروژه دارد از بخش‌های مختلف استفاده می‌کند. برای این منظور کافی است تنظیمات مورد نیاز در بخش مورد نظر را در حالت فعال قرار دهد. با انتخاب یک میکروکنترلر مناسب می‌توانیم پروژه‌های الکترونیکی ساده را با صرفه‌جویی در تعداد آی‌سی مدارهای دیجیتال اجرا کنیم. مزیت این روش سرعت بالای مدار و مقرون به صرفه بودن آن است.

میکروکنترلرهای AVR در دسته کوچک (Tiny)، متوسط (s ۹۰)، بزرگ (Mega) و خیلی بزرگ (Xmega) دسته‌بندی می‌شوند. با توجه به مدل‌های متداول موجود در بازار ایران، یک نمونه آی‌سی از سری Mega به نام ATmega8 را معرفی می‌کنیم. این میکروکنترلر در سه نوع بسته‌بندی وجود دارد. دو مدل آن از نوع SMD و نوع ساده آن با بسته‌بندی دو ردیفه (DIP) است. در شکل ۹-۶ دو نوع بسته‌بندی DIP و SMD را مشاهده می‌کنید. میکروکنترلر از نوع DIP را بهتر می‌توانیم روی بردبرد قرار دهیم و پروژه را اجرا کنیم. بنابراین عملاً از این نوع آی‌سی‌های میکروکنترلر بیشتر استفاده می‌شود.



شکل ۹-۶ آرایش پایه‌های میکروکنترلر ATmega8 از نوع DIP و SMD

استخراج و ترجمه اطلاعات فنی بخش‌های مختلف میکروکنترلر

هدف: کسب مهارت در ارتباط با استخراج اطلاعات مختلف از برگه اطلاعات میکروکنترلر ATmega ۸.

مواد، ابزار و تجهیزات: لوازم التحریر - فرهنگ لغات - برگه اطلاعات خانواده میکروکنترلر AVR شکل ۶-۶

مراحل اجرای کار

۱ با مراجعه به برگه اطلاعات میکروکنترلر ATmega ۸ مشخصات کلی آن را بررسی و جدول ۴-۶ را کامل کنید.

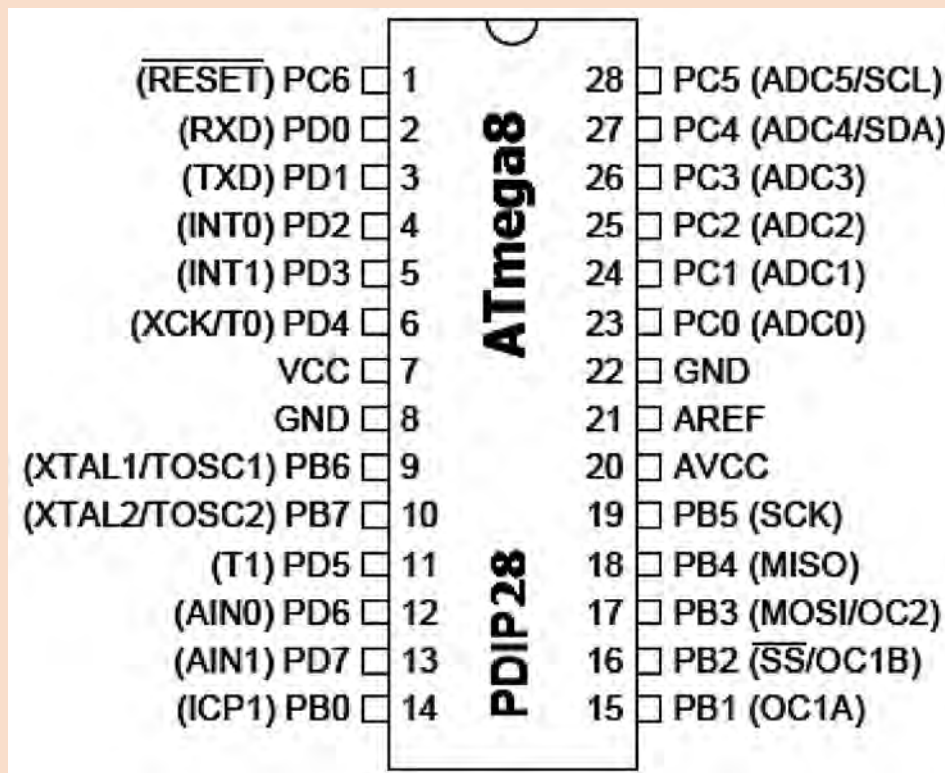
کار عملی ۲



جدول ۶-۴

ردیف	موضوع	پاسخ
۱	تعداد پایه‌های میکروکنترلر	
۲	تعداد پایه‌های ورودی و خروجی دیجیتال	
۳	تعداد تایمر	
۴	سرعت (پالس ساعت - کلاک پالس)	
۵	ولتاژ کار	
۶	تعداد ورودی‌های آنالوگ به دیجیتال	
۷	مقدار جریان مصرفی میکروکنترلر	

۲ شماره پایه‌های میکروکنترلر ۸ ATmega نشان داده شده در شکل ۶-۱۰ را، در جدول ۶-۵ بنویسید.



شکل ۶-۱۰ پایه‌های میکروکنترلر ۸ ATmega

جدول ۵-۶

شماره پایه	نام پایه	شماره پایه	نام پایه	شماره پایه	نام پایه	شماره پایه	نام پایه
	ADC ^۰		PORTD ^۰		PORTC ^۰		PORTB ^۰
	ADC ^۱		PORTD ^۱		PORTC ^۱		PORTB ^۱
	ADC ^۲		PORTD ^۲		PORTC ^۲		PORTB ^۲
	ADC ^۳		PORTD ^۳		PORTC ^۳		PORTB ^۳
	ADC ^۴		PORTD ^۴		PORTC ^۴		PORTB ^۴
	ADC ^۵		PORTD ^۵		PORTC ^۵		PORTB ^۵
	AIN ^۰		PORTD ^۶		PORTC ^۶		PORTB ^۶
	AIN ^۱		PORTD ^۷		PORTC ^۷		PORTD ^۷

چگونه یک پایه هم می‌تواند در نقش ADC^۱ آنالوگ و هم PORTC^۱ دیجیتال کار کند؟ یا پایه RXD (دریافت‌کننده اطلاعات سری) تبدیل به PORTD^۰ شود؟

فکر کنید



با مراجعه به رسانه‌های مختلف ملاک‌های انتخاب میکروکنترلر را برای کنترل یک سامانه مثلاً روشنایی ساختمان پیدا کنید و نتیجه را در قالب یک گزارش به کارگاه ارائه دهید.

پژوهش



۶-۶ برنامه‌نویسی در میکروکنترلرها

میکروکنترلر یک قطعه قابل برنامه‌ریزی است. یعنی می‌توان روش کار را با توجه به نیاز برای این قطعه در قالب یک برنامه تعریف کرد. برای اینکه بتوانیم ایده و روش اجرای کار را برای میکروکنترلر به‌گونه‌ای تعریف کنیم که بتواند فرایند کار را به‌طور دقیق اجرا کند، به عبارت دیگر باید با میکروکنترلر سخن بگوییم و مراحل اجرای کار را با نوشتن دستورها به ترتیب به میکروکنترلر معرفی کنیم. نوشتن دستورها با قواعد خاص تعیین شده را برنامه‌نویسی می‌نامند. برای سخن گفتن از طریق متن، حروف و نمادها با میکروکنترلر نیاز به حروف، کلمات و قواعد دستوری خاص داریم که به آن زبان نوشتاری می‌گویند. برای برنامه‌نویسی میکروکنترلرها می‌توانیم از زبان‌های اسمبلی، زبان بیسیک یا زبان C استفاده کنیم.

با مراجعه به رسانه‌های مختلف جست‌وجو کنید و تعداد زبان‌های موجود برای میکروکنترلرها مانند زبان ماشین را پیدا کنید. کدام یک از این زبان‌ها کاربردی‌تر است؟ چرا؟ نتیجه را در قالب یک گزارش به کارگاه ارائه دهید.

پژوهش



۷-۶ انواع زبان برنامه‌نویسی

■ زبان برنامه‌نویسی اسمبلی

در این زبان مجموعه‌ای از دستورها به صورت کلمات انگلیسی مخفف شده هستند و هر دستور در یک خط نوشته می‌شود.

مثال

دستور ۵, R۳ LD (بخوانید عدد ۵ را در R۳ لود کنید).

دستور (LD R۳) مخفف کلمات "Load Register R۳" است و به معنی و مفهوم عدد ۵ در رجیستر R۳ قرار گیرد.

مثال

دستور ۱۰۲ JZ

(JZ ۱۰۲) به مفهوم، "If Zero Then Jump To Line ۱۰۲" است. یعنی اگر نتیجه دستور قبل صفر شده است به خط ۱۰۲ پرش کن.

آموزش زبان برنامه (اسمبلی) به دلیل مخفف بودن، دور بودن از زبان رایج انسان‌ها و نیاز به آشنایی با جزئیات سخت‌افزار میکروکنترلر، ساده نیست. اما برنامه نوشته شده با اسمبلی می‌تواند کم‌حجم باشد، حافظه کمتری نیاز داشته باشد و سریع‌تر از زبان‌های دیگر اجرا شود.

■ زبان برنامه‌نویسی بیسیک Basic

این زبان بر مبنای اصول نوشتاری ساده زبان انگلیسی (البته کلمات منطقی) طرح شده است. سادگی آن باعث شده اکثر دانش‌آموزان و علاقه‌مندان به کار با میکروکنترلر را به خود جلب کند.

■ زبان برنامه‌نویسی C

زبان برنامه‌نویسی C مانند زبان بیسیک از کلمات منطقی انگلیسی و همچنین قواعد نوشتاری خاصی پیروی می‌کند. به دلیل حجم نسبتاً کم فایل تولید شده و سرعت مناسب دستوراتی که ارائه می‌کند، طرفداران بی‌شماری در سراسر جهان دارد. امروزه برای انواع میکروکنترلرها از خانواده‌های مختلف، کامپیوترهای کوچک و بزرگ از زبان C استفاده می‌شود. مزیت استفاده از زبان C این است که در یک خانواده میکروکنترلر، به راحتی می‌توان برنامه‌های یک میکروکنترلر را با کپی کردن به میکروکنترلر دیگری انتقال داد. برای سایر خانواده‌های میکروکنترلر نیز می‌توان با تغییر بعضی از کلید واژه‌ها از برنامه داخلی آنها استفاده کرد.

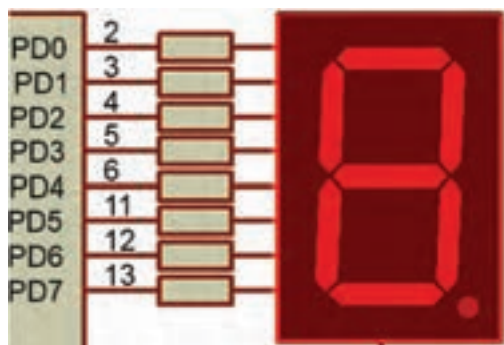
درمورد ویژگی‌های زبان برنامه‌نویسی بیسیک و زبان C تحقیق کنید و نتایج را در کارگاه ارائه نمایید.

پژوهش

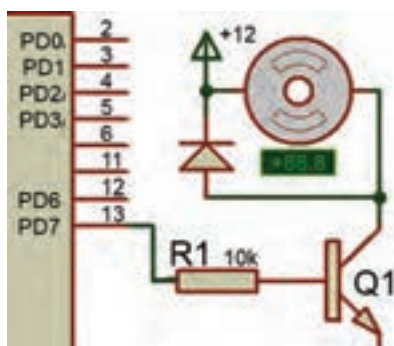


۸-۶ مراحل اجرای یک پروژه با میکروکنترلر

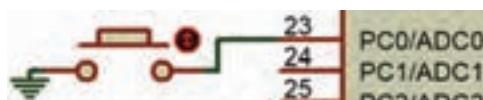
برای اجرای یک پروژه با میکروکنترلر صرف نظر از نوع میکروکنترلر و زبان برنامه‌نویسی، باید میکروکنترلر را به شرح زیر مشخص کنیم:



الف) ۸ پایه برای فرمان به هفت قطعه‌ای

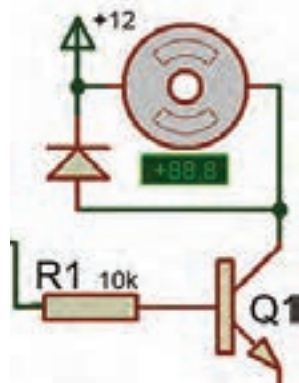


ب) یک پایه برای فرمان به ترانزیستور



پ) پایه ورودی برای اتصال یک کلید

شکل ۸-۱۱ پایه‌های میکروکنترلر جهت اتصال به قطعات جانبی



شکل ۸-۱۳ نقشه یک نمونه مدار جانبی

تعداد ورودی و خروجی (دیجیتال و آنالوگ)

مورد نیاز:

تعداد پایه ورودی و خروجی موردنیاز با توجه به پایه‌های قطعات جانبی میکروکنترلر تعیین می‌شود. همان‌طور که در شکل ۸-۱۱ مشاهده می‌کنید. برای هر کلید یک بیت ورودی و برای هر LED یا رله، یک بیت خروجی در نظر گرفته می‌شود. برای راه‌اندازی نمایشگر هفت قطعه‌ای (۷ SEG) نیاز به ۸ بیت (۱ Byte) داریم، زیرا عملاً باید ۸ پایه استفاده شود.

پس از اختصاص دادن پایه‌ها، تعداد پایه‌ها را با هم جمع می‌کنند به این ترتیب تعداد پایه‌های مورد نیاز مشخص می‌شود. از روی تعداد پایه‌ها و نیازهای فنی میکروکنترلر را انتخاب می‌کنیم.

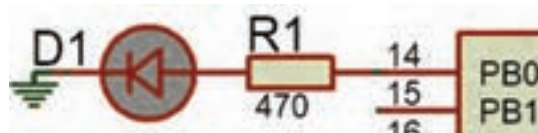
رسم نقشه فنی مدار میکروکنترلر در

نرم‌افزار پروتئوس (Proteus)

در رسم نقشه فنی مدارهای میکروکنترلر آدرس‌های ورودی و خروجی برای محیط برنامه‌نویسی مشخص می‌شوند. در شکل‌های ۸-۱۱ و ۸-۱۲ اتصال قطعات جانبی به پایه‌های میکروکنترلر نشان داده شده است.

بستن مدار جانبی میکروکنترلر

برای اطمینان از صحت عملکرد مدار، مدارهای الکترونیکی مرتبط با میکروکنترلر را طبق شکل ۸-۱۳ روی برد می‌بندند. این مدارها معمولاً به صورت مستقل در روی برد برد بسته می‌شوند که میکروکنترلرها را با حسگرها، راه‌انداز موتورها یا نمایشگرها ارتباط می‌دهد.



شکل ۸-۱۲ اختصاص پایه PB0 به LED

✓ نوشتن برنامه دستوره‌های اجرای میکروکنترلر با یک زبان برنامه نویسی

برنامه نویسی با یک نرم افزار مستقل انجام می شود، مثلاً برای نوشتن برنامه با زبان C از نرم افزار CodeVision و برای نوشتن زبان بیسیک از نرم افزار Bascom استفاده می شود.

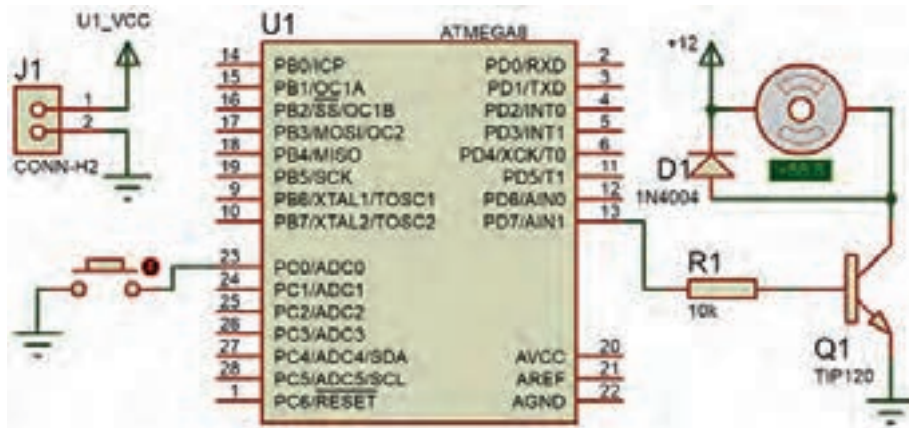
فیلم نصب نرم افزارهای ۷۳،۱۲ AVR CodeVision و Bascom (Basavr) را ببینید.

فیلم



✓ شبیه سازی مدار و اجرای برنامه در نرم افزار پروتئوس

نرم افزار پروتئوس Proteus برای شبیه سازی مدارهای الکترونیکی، دیجیتال و به خصوص میکروکنترلر AVR توانایی بالایی دارد. روش کار با نرم افزار پروتئوس را با مراجعه به منابع مختلف مطالعه کنید، سپس بالاترین نسخه نرم افزار را نصب کنید و چند نمونه مدار را ببینید تا مهارت لازم را به دست آورید. شکل ۱۴-۶ یک نمونه مدار شبیه سازی شده را نشان می دهد.



شکل ۱۴-۶ مدار شبیه سازی شده در نرم افزار Proteus

فیلم نصب نرم افزار Proteus را ببینید.

فیلم



نرم افزار Proteus را بر روی رایانه نصب کنید. نحوه نصب نرم افزار و مراحل اجرایی آن را در کارگاه ارائه دهید.

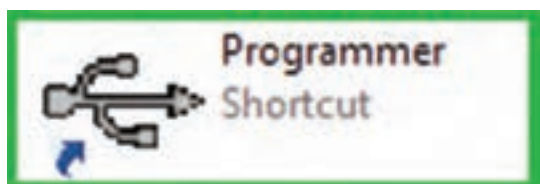
پژوهش



در هنگام شبیه سازی با نرم افزار، دستوره‌های تعریف شده برای میکروکنترلر مورد آزمایش قرار می گیرد، در صورتی که مرحله‌ای از دستورها به طور صحیح اجرا نشود، یا خطایی در اجرای برنامه اعلام شود با مراجعه به نرم افزار برنامه نویسی برنامه را اصلاح و دوباره بارگذاری و اجرا می کنیم، تا نتیجه مطلوب حاصل شود شکل ۱۵-۶. در اجرای پروژه‌ها ممکن است این فرایند چندین بار تکرار شود تا خطاها برطرف و مدار پاسخ دهد.



شکل ۱۵-۶ جابه‌جایی بین دو نرم‌افزار برای اصلاح برنامه و اجرای پروژه



شکل ۱۶-۶ نماد نرم‌افزار یک نمونه پروگرامر

✓ پروگرام کردن میکروکنترلر

بارگذاری یا انتقال برنامه به میکروکنترلر به وسیله مدار پروگرامر (Programmer) مستقل و سازگار با نرم‌افزار برنامه‌نویسی انجام می‌شود. مدار پروگرامر و نرم‌افزار مرتبط به صورت آماده در بازار ارائه می‌شود، شکل ۱۶-۶. همچنین خودتان می‌توانید این مدار را بسازید.

به کمک مربی خود یک مدار پروگرامر طراحی کنید. مدار سخت‌افزاری آن را بسازید، و نتیجه را به کارگاه ارائه دهید.

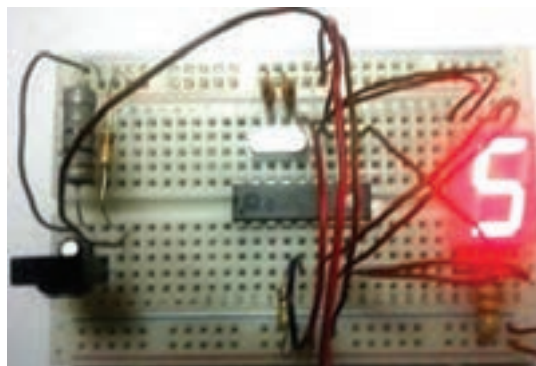
فعالیت



✓ بستن مدار سخت‌افزاری پروژه میکروکنترلر



ب) برد مدار چاپی پروژه میکروکنترلر



الف) مدار سخت‌افزاری روی برد

شکل ۱۷-۶ مدارهای سخت‌افزاری پروژه با میکروکنترلر

مدار سخت‌افزاری میکروکنترلر را با قطعات جانبی آن روی برد می‌بندیم، بعد از مطمئن شدن از صحیح کار کردن مدار، فیبر مدار چاپی آن را طراحی کرده و قطعات را بر روی برد فیبر مدار چاپی مونتاژ و راه‌اندازی می‌کنیم، شکل ۱۷-۶.

۹-۶ ساختار برنامه‌نویسی به زبان C



#include (هدر فایل)

ماکروها

متغیرهای عمومی

توابع

void main (void)

{

....

....

....

برنامه

....

....

}

در این قسمت قصد داریم برنامه‌نویسی میکروکنترلر را با زبان C و نرم‌افزار Codevision انجام دهیم، این زبان برنامه‌نویسی کاربرد زیادی دارد که در آینده با دستورهای آن آشنا خواهید شد. ساختار برنامه C به شکل ۶-۱۸ است.

در زبان برنامه‌نویسی دستورات بر سه نوع هستند:

✓ دستورهایی که به آنها دستورهای استاندارد C گفته می‌شود.

✓ دستورهایی که توسط فایل‌های جانبی به نام کتابخانه (Header فایل‌ها) به برنامه اضافه می‌شوند مانند فایل (delay.h) که در بالای برنامه توسط کد دستوری `#include <delay.h>` به برنامه اضافه می‌شود. این دستور با شماره ۱ در جدول ۶-۶ نشان داده شده است.

✓ به دستورهایی که کاربر برنامه‌نویس، برای اجرای یک عمل در درون برنامه ایجاد می‌کند توابع می‌گویند.

شکل ۶-۱۸ نرم‌افزار CodeVision و ساختار آن

در زبان C، یک برنامه به نام مفسر یا مترجم کلی

وجود دارد که برنامه نوشته شده را به زبان ماشین (کُد) ترجمه می‌کند. در این فرایند فایل ساخته شده توسط مفسر در مقایسه با فایل نوشته شده توسط کاربر کوچک‌تر شده و سریع‌تر عمل می‌کند.

■ قواعد ساده نوشتاری در زبان برنامه‌نویسی C

✓ یک بلوک برنامه یا تابع یا فرایند مربوط به دستورات شرطی با علامت "}" شروع و در انتها با علامت "{" به پایان می‌رسد. علامت "}" با شماره ۳ و علامت "{" با شماره ۱۱ در جدول ۶-۶ نشان داده شده است.

✓ برنامه تابع اصلی پس از مجموعه کلمه‌های Void main (void) و علامت "}" قرار می‌گیرد، (شماره ۲ در جدول ۶-۶). در ادامه دستورهای لازم برای تنظیمات اولیه پایه‌ها و سخت‌افزارهای درون میکروکنترلر به اجرا در می‌آید (شماره ۴ در جدول ۶-۶).

✓ برنامه مورد نظر ما بعد از خط `while(۱){` شروع می‌شود و باید تا قبل از علامت "}" پایان یابد. این برنامه به دلیل نوشتن عدد یک در دستور `while(۱){` می‌تواند تا بی‌نهایت ادامه یابد و اجرا شود (شماره ۵ در جدول ۶-۶).

✓ اگر توجه کنیم در انتها دو بار علامت "}" وجود دارد که یکی مربوط به پایان محدوده `while(۱){` و دیگری پایان محدوده `Void main (void){` یا همان پایان نهایی است (شماره ۱۰ و ۱۱ در جدول ۶-۶).

✓ در انتهای هر عبارت و در پایان هر خط باید علامت (;) قرار گیرد تا مفسر برنامه بتواند برنامه را به درستی تفسیر کند.

✓ زبان C به حروف بزرگ و کوچک حساس است. لذا اگر دستور یا متغیر حروف کوچک باشد، اگر به اشتباه حتی یکی از حروف بزرگ نوشته شود مفسر روی این کلمه اعلام خطا می‌کند. لذا در نوشتن برنامه حتماً به حروف بزرگ و کوچک اختصاص داده شده به عبارات توجه کنید.

مثال ۳

عبارت درست ;PORTB.۱=۰

عبارت نادرست ;portb.۱=۰

جدول ۶-۶ ساختار یک برنامه ساده

شماره	توضیح	دستور
۱	شروع برنامه با این دستورها	#include <mega.h> #include <delay.h>
۲	تابع اصلی برنامه	void main(void)
۳	علامت شروع برنامه	{
۴	اجرای دستورهای لازم برای تنظیمات اولیه پایه‌ها و سخت‌افزارهای درون میکروکنترلر	DDRB.۲=۱; ...
۵	حلقه بی‌نهایت برنامه	While(۱)
۶	شروع حلقه While	{
۷	مقداردهی بیت ۲ از پورت B با مقدار ۱	PORTB.۲=۱;
۸	دستور تأخیر در روند اجرا	delay_ms(۵۰۰);
۹	مقداردهی بیت ۲ از پورت B با مقدار ۰	PORTB.۲=۰;
۱۰	پایان محدوده { while(۱) بعد از اجرای تأخیر	delay_ms(۵۰۰); }
۱۱	پایان محدوده برنامه اصلی	}

✓ هر پرانتز باز شده " (" باید با یک پرانتز بسته ") " و هر علامت " { " باز شده باید با یک علامت " } " بسته شده کامل شود.

فیلم اجرای مدار چشمک‌زن با نرم‌افزار Basavr را در کلاس ببینید.

فیلم



فیلم اجرای مدار چشمک‌زن با نرم‌افزار Code Vision را در کلاس ببینید. در کارگاه در مورد دستورات و قواعد این دو زبان بحث کنید.

فیلم



در کلاس در مورد دستورها و قواعد این دو زبان بحث کنید و نتیجه را در یک جدول به صورت خلاصه بنویسید و به کارگاه ارائه دهید.

کار گروهی



همان‌طور که در فیلم مشاهده کردید در داخل نرم‌افزار کدویژن یک برنامه هوشمند به نام Wizard وجود دارد که نماد آن شبیه کلاه جادوگری است. این برنامه کارهای اولیه (مانند تنظیم اولیه پایه‌ها و سخت‌افزارهای درون میکروکنترلر) را به صورت پرسش و پاسخ و یا به صورت پنجره‌ای در اختیار برنامه‌نویس قرار می‌دهد. بعد از تنظیم، موارد مختلف تعریف شده به وسیله برنامه نوشته می‌شود و برنامه‌نویس فقط برنامه خود را در بین عبارات "{(۱)while}" و... "}" قرار می‌دهد.

نکته



۱۰-۶ کاربرد دستورهایی نوشته شده در زبان C

■ کار با پایه‌ها و پورت‌ها و آشنایی با متغیرهای Bit، Byte

✓ برای روشن و خاموش کردن یک پایه خروجی میکروکنترلر، نیاز به انتقال یک بیت "۱" یا "۰" منطقی است.

دستور $PORTD.X=1$ مفهوم یک بیت را دارد. X شماره یکی از پایه‌های پورت D است، که ارزش منطقی ۱ به آن داده می‌شود.

مثال ۴

دستور $PORTD.4=1$ را شرح دهید.

پاسخ

ارزش منطقی پایه شماره ۴ در پورت D میکروکنترلر مورد آزمایش برابر با "۱" است.

مثال ۵

دستور $PORTB.1=0$ را شرح دهید.

پاسخ

ارزش منطقی پایه شماره ۱ در پورت B میکروکنترلر مورد آزمایش برابر با "۰" است.

✓ برای اختصاص دادن ارزش منطقی "۱" یا "۰" به یک متغیر منطقی از یک پایه ورودی میکروکنترلر، از دستور $A=PIND.2$ استفاده می‌شود. این دستور می‌گوید ارزش منطقی متغیر A را از پایه شماره ۲ پورت D بخوان.

به متغیری که ارزش آن ۱ یا ۰ است. بیت bit یا bool می‌گویند.

نکته



سؤال

دستور $X = \text{PINC}$ چه عملی را انجام می‌دهد؟
اگر بخواهیم تمام پایه‌های یک پورت (مانند پورت B)، ارزش منطقی " ۱ " را داشته باشند، یعنی فعال شوند، دستور $\text{PORTB} = 255$ را به کار می‌بریم.

مثال ۶

دستور $\text{PORTB} = 255$ را توضیح دهید.

پاسخ

با اجرای این دستور به تمامی پایه‌های پورت B یعنی $\text{PB}0$ تا $\text{PB}7$ فرمان یک یا روشن می‌رسد و قطعاتی که به آن اتصال دارند فعال می‌شوند.

(مثل $\text{PD}0 \dots \text{PD}7$) باشد همان دستور قبل با شکلی ساده‌تر استفاده می‌شود. مانند: $\text{PORTB} = 255$ که تمامی پایه‌ها پورت B تا فرمان یک یا روشن می‌رسد و قطعاتی که به آن اتصال دارند فعال می‌شوند.

سؤال

عدد ۲۵۵ در سیستم باینری برابر چه عددی است؟

$0xd3$ نوشتن عدد در قالب هگزادسیمال است. اگر این دستور به کار رود $\text{PORTD} = 0xd3$ ، کدام پایه‌ها روشن و کدام پایه‌ها خاموش می‌شوند؟ نتیجه را به کارگاه ارائه دهید.

تحقیق کنید



جدولی تهیه کنید و دستورهای پرکاربرد را در یک ستون آن بیاورید و در ستونی دیگر مفهوم و عملکرد دستور را بنویسید.

فعالیت



✓ برای خواندن تمامی پایه‌های یک پورت از دستور $Z = \text{PINC}$ استفاده می‌شود. Z یک متغیر است که اندازه آن ۸ بیت برابر با تعداد پایه‌های پورت C است. به متغیری که اندازه آن ۸ بیت باشد Byte می‌گویند. در زبان برنامه‌نویسی C این نوع متغیر را با نام کاراکتر (char یا unsigned char) می‌شناسند.

■ دستور تأخیر، شرط، حلقه و تکرار

✓ **دستور تأخیر** : delay_ms (۲۰۰) یا delay_us (۳۵۰) این دو دستور برای ایجاد تأخیر در روند برنامه به کار می‌رود و هر کدام دارای مفهومی به شرح زیر است:

تأخیری برابر ۲۰۰ میلی ثانیه delay_ms (۲۰۰)

تأخیری برابر ۳۵۰ میکرو ثانیه delay_us (۳۵۰)

✓ **دستور شرط** : $\{ \dots \} \text{if}(A > B)$ این دستور این گونه عمل می‌کند که اگر A بزرگ‌تر یا مساوی B باشد. دستورهای را که بین $\{ \}$ قرار دارد، فقط یکبار اجرا می‌کند و اگر شرط برقرار نباشد به ادامه برنامه می‌رود.

✓ **دستور حلقه**: دو دستور پرکاربرد به قرار زیر است.

الف) $\{ \dots \} \text{while}(a == x)$ این فرمان تا زمانی که متغیر a با متغیر x برابر باشد C دستورهای که مابین $\{ \}$ قرار دارند را مرتباً اجرا می‌کند. اگر a با x برابر نباشد از حلقه تکرار خارج شده به ادامه برنامه می‌رود.

ب) $\{ \dots \} \text{for}(i = 0; i < 10; i++)$ این فرمان ۱۰ بار دستورهای مابین علامت $\{ \}$ را اجرا می‌کند و بعد از آن برنامه را ادامه می‌دهد.

i++ چه عملی انجام می دهد؟ دستور مشابه آن چیست؟ نتیجه را به کارگاه ارائه دهید.

جست و جو کنید



توجه



کار عملی ۳



دستورهای شرح داده از پرکاربردترین دستورهای زبان C هستند. سایر دستورها در کتاب همراه هنرجو آمده است و همچنین برخی از دستورهای دیگر را در روند اجرای برنامه های مدارهای کاربردی با میکروکنترلر (واحد یادگیری ۷) فرا می گیرید.

دستورنویسی ساده با زبان C و بستن مدارهای جانبی میکروکنترلر

هدف: کسب مهارت در نوشتن دستورهای ساده برای میکروکنترلر ۸ ATmega و راه اندازی مدارهای جانبی مواد، ابزار و تجهیزات: لوازم التحریر - دستورهای زبان C - برگه اطلاعات میکروکنترلر ۸ ATmega - رایانه - نرم افزار پروتئوس Proteus - ترانزیستور BC ۱۴۰ - دیود ۱N۴۰۰۱ - مقاومت ۱ KΩ یک چهارم وات - لامپ ۱۲ ولت - رله ۱۲ ولتی یک کنتاکت - ترانسفورماتور ۶۷×۲/۲۲۰ - موتور ۱۲ ولتی کوچک و کم وات مراحل اجرای کار

۱ دستور بنویسید که پین ۲ پورت C را روشن یا وضعیت منطقی آن را "۱" کند.

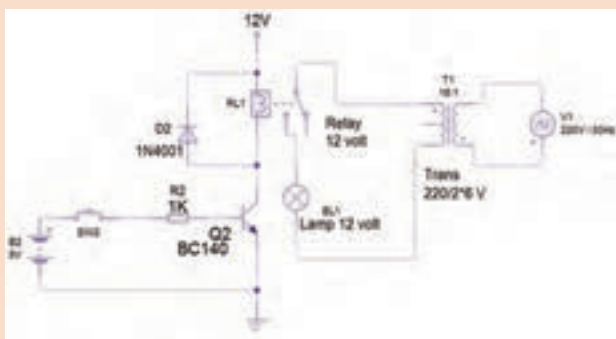
دستور	شماره پین ۲ پورت C	

۲ دستور بنویسید که تمام پین های پورت D را روشن و خاموش کند.

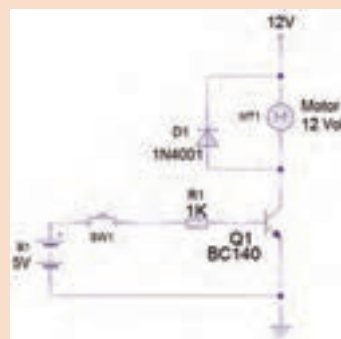
۳ دستوری برای تأخیر ۳۰۰ میلی ثانیه در روند اجرای برنامه بنویسید.

۴ برای روشن و خاموش کردن کامل یک پورت (مثلاً پورت B) از چه دستوری استفاده می کنیم؟

۵ مدارهای شکل ۶-۱۹ و ۶-۲۰ قرار است با یک میکروکنترلر مناسب کنترل شود، هریک از مدارها را در نرم‌افزار پروتئوس Proteus ببندید و آنها را شبیه‌سازی کنید.



شکل ۶-۲۰ مدار کنترل روشنایی لامپ با ولتاژ AC



شکل ۶-۱۹ مدار راه‌انداز (Driver)

۶ مدار شکل ۶-۱۹ را بر روی بردبرد ببندید. عملکرد مدار را به اختصار توضیح دهید.

۷ مدار شکل ۶-۲۰ را بر روی بردبرد ببندید. نحوه راه‌اندازی مدار را به اختصار توضیح دهید.

۸ به نظر شما راه‌اندازی مدارهای شکل ۶-۱۹ و ۶-۲۰، با یک میکروکنترلر ۸ ATmega چگونه است؟ توضیح دهید. با راهنمایی معلم نقشه مدار را در شکل ۶-۲۱ و ۶-۲۲ رسم کنید.



شکل ۶-۲۱ ترسیم نقشه شکل ۶-۱۹ با ATMEGA ۸



شکل ۶-۲۲ ترسیم نقشه شکل ۶-۲۰ با ATMEGA ۸

الگوی آزمون نظری پایان واحد یادگیری:

۱. کوچک ترین واحد حافظه بایت است. ☐ درست ☐ نادرست
۲. CPU و واحد ورودی و خروجی بخشی از است.
۳. USB مربوط به کدام بخش یک سیستم کامپیوتری یا میکروکنترلر می باشد؟
۴. Counter در میکروکنترلرها برای چه کاربردی استفاده می شود؟
۵. خلاصه ای از برگه اطلاعات مشخصات یک میکروکنترلر در شکل ۶-۲۳ نشان داده شده است، به سؤال های مرتبط به آن پاسخ دهید.

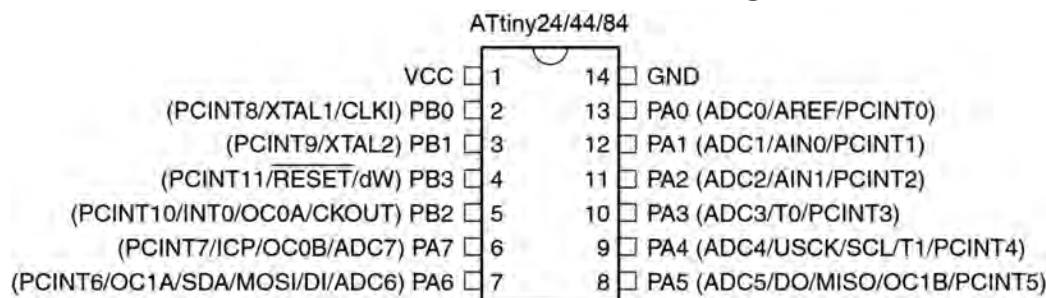
16 MHz STM8S 8-bit MCU, 8 Kbytes Flash, 128 bytes data EEPROM, 10-bit ADC, 3 timers, UART, SPI, I²C

شکل ۶-۲۳

- ✓ مورد شماره ۱ چه پارامتری از میکروکنترلر را بیان می کند؟
 - الف) سرعت واقعی cpu برحسب فرکانس
 - ب) سرعت cpu بر مبنای تعداد دستور در ثانیه
 - ج) اندازه محاسبات بر مبنای بیت
 - د) یکی از پارامترهای سرعت بر مبنای فرکانس
- ✓ مورد شماره ۲ کدام پارامتر از میکروکنترلر را بیان می کند؟
 - الف) سرعت cpu برحسب فرکانس
 - ب) تعداد هسته cpu
 - ج) اندازه محاسبات بر مبنای بیت
 - د) اندازه واحد حافظه داخلی بر مبنای بیت
- ✓ شماره ۳ به کدام حافظه اشاره می کند، تفاوت آن با مورد شماره ۴ کدام است؟
 - الف) خواندنی و نوشتنی - فقط خواندنی ب) فقط خواندنی - خواندنی نوشتنی
 - ج) حافظه کمکی - اصلی
 - د) هر دو حافظه فقط خواندنی، ولی تفاوت در نحوه دسترسی به آنها
- ✓ کدام گزینه کاربرد شماره ۵ را مشخص می کند؟
 - الف) اندازه گیری سنسورها
 - ب) شمارش بسته ها
 - ج) ارتباط سریال
 - د) تکرار برنامه ها با زمان مشخص
- ✓ اطلاعات کدام حافظه با قطع تغذیه میکروکنترلر پاک می شود؟
 - الف) RAM
 - ب) ROM
 - ج) EPROM
 - د) حافظه فلش
- ✓ هر یک از کاربردهای شماره ۶ شامل (UART, SPI, I²C) در میکروکنترلر را شرح دهید.
 - الف) UART
 - ب) SPI
 - ج) I²C
- ✓ شماره ۷ برای چه کاربردی در میکروکنترلر قرار گرفته است؟ اگر مقدار آن ۸bit باشد چه تفاوتی در کاربرد آن ایجاد می شود؟

آزمون عملی پایان واحد یادگیری:

الگوی آزمون ۱: یکی از راه‌های به دست آوردن اطلاعات میکروکنترلر مشاهده شکل پایه‌های (Pin Config) آن است. در شکل ۶-۲۴ پایه‌های یک میکروکنترلر معرفی شده است. با توجه به اطلاعات روی شکل به سؤالات زیر پاسخ دهید؟



شکل ۶-۲۴ میکروکنترلر Atiny AVR

- ۱ تعداد پایه‌های قابل استفاده به عنوان ورودی و خروجی دیجیتال چند پایه است؟
 - ۲ پایه‌های ورودی آنالوگ به دیجیتال کدام پایه‌ها هستند؟
 - ۳ آیا می‌توان اسیلاتور داخلی با کریستال خارجی میکروکنترلر را فعال کرد؟ از کدام پایه‌ها استفاده می‌شود؟
 - ۴ کدام پایه‌های میکروکنترلر مقایسه کننده آنالوگ است؟
 - ۵ مفهوم پایه ۱۳ میکروکنترلر (AREF) شکل ۶-۲۴ را مشخص کنید؟
 - ۶ این میکروکنترلر چند تایمر دارد؟
 - ۷ پایه‌هایی که برای پروگرام کردن میکروکنترلر استفاده می‌شوند کدام هستند؟
 - ۸ این میکروکنترلر برای کدام پروژه‌های زیر مناسب است؟ چرا؟
- الف) ۴ عدد کلید - یک سون سگمنت کامل (۷قطعه + LED ممیز) به عنوان نمایش دهنده و سه سنسور برای اندازه گیری گرما و فشار مخزن
- ب) ۳ عدد کلید، ۵ عدد LED و ۴ سنسور برای اندازه گیری دما و فشار یک مخزن
- الگوی آزمون ۲:** برای کنترل یک مدار چشمک زن با دو دیود نورانی قرمز و سبز یک میکروکنترلر ۸ ATMEGA نیاز است. می‌خواهیم با فشردن یک کلید فشاری شروع به کار کند و با فشردن کلید فشاری دوم مدار چشمک زن متوقف شود. نقشه مدار را با نرم افزار پروتئوس شبیه سازی کنید.

ارزشیابی شایستگی

شرح کار:

- ۱ شرح عملکرد کلی میکروکنترلرها و میکروپروسسورها و مقایسه آنها با یکدیگر
- ۲ شرح قابلیت‌های میکروکنترلرهای AVR - چگونگی برنامه‌نویسی به زبان C و نوشتن یک برنامه ساده
- ۳ نصب و راه‌اندازی برنامه پروتئوس - شبیه‌سازی یک مدار در نرم‌افزار

استاندارد عملکرد:

میکروکنترلر و چگونگی برنامه‌ریزی آن با رعایت استاندارد تعریف شده

شاخص‌ها:

- ۱ راه‌اندازی نرم‌افزار پروتئوس (۵ دقیقه)
- ۲ تشریح عملکرد مدارهای جانبی میکروکنترلر (۱۵ دقیقه)
- ۳ نوشتن برنامه ساده به زبان C (۱۰ دقیقه)
- ۴ شبیه‌سازی یک نمونه مدار با نرم‌افزار (۱۵ دقیقه)

شرایط انجام کار و ابزار و تجهیزات:

شرایط: مکان انجام کار با کف عایق یا آنتی استاتیک - نور مناسب برای کارهای مطالعه - میز کار استاندارد برای رایانه - رایانه متصل به شبکه اینترنت - انجام کار در حال نشسته.

ابزار و تجهیزات: لوازم التحریر - دستورهای زبان C - برگه اطلاعات میکروکنترلر ۸ ATmega - رایانه - نرم‌افزار پروتئوس Proteus - ترانزیستور BC ۱۴۰ - دیود ۱N۴۰۰۱ - مقاومت ۱ K Ω یک چهارم وات - لامپ ۱۲ ولت - رله ۱۲ ولتی یک کنتاکت - ترانسفورماتور ۲۲۰/۲×۶V - موتور ۱۲ ولتی کوچک و کم وات

معیار شایستگی:

ردیف	مراحل کار	حداقل نمره قبولی از ۳	نمره هنرجو
۱	راه‌اندازی نرم‌افزار پروتئوس	۱	
۲	تشریح عملکرد مدارهای جانبی میکروکنترلر	۲	
۳	نوشتن برنامه ساده به زبان C	۲	
۴	شبیه‌سازی یک نمونه مدار با نرم‌افزار	۲	
	شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست‌محیطی و نگرش: ۱- رعایت نکات ایمنی دستگاه‌ها ۲- دقت و تمرکز در اجرای کار ۳- شایستگی تفکر و یادگیری مادام‌العمر ۴- اخلاق حرفه‌ای	۲	
میانگین نمرات			*

* حداقل میانگین نمرات هنرجو برای قبولی و کسب شایستگی ۲ می‌باشد.

واحد یادگیری ۷

مدارهای کاربردی با میکروکنترلر

آیا تا به حال فکر کرده‌اید

- چه تعداد ورودی و خروجی قابل اتصال به میکروکنترلر است؟
- چه نوع سنسورهایی را می‌توان به میکروکنترلر متصل کرد؟
- آیا می‌توان مدارهای جریان بالا را به میکروکنترلر متصل کرد؟
- چه نوع سخت افزارهایی را می‌توان به میکروکنترلر اتصال داد؟
- چگونه می‌توان یک دستگاه پروگرامر ساخت؟
- چگونه می‌توان سرعت و جهت چرخش (چپ گرد و راستگرد) یک موتور DC را تغییر داد؟
- آیا در لوازم خانگی میکروکنترلر به کار می‌رود؟

یک میکروکنترلر بدون برنامه‌ریزی شدن، قابلیت کاربردی ندارد، با برنامه‌ریزی، میکروکنترلر توانایی کنترل مدارهای الکترونیکی را پیدا می‌کند. در واقع برنامه هر میکروکنترلر روح و جانی است که در آن دمیده می‌شود و آن را پویا و فعال می‌کند. برنامه‌ریزی میکروکنترلر را با استفاده از یک نرم‌افزار به نام مترجم (Compiler) انجام می‌دهند. این برنامه را به وسیله دستگاهی به نام پروگرامر در میکروکنترلر بارگذاری (Program) می‌کنند. پروگرامر یک واسطه الکترونیکی ارتباطی بین کامپیوتر و میکروکنترلر است. شما به وسیله میکروکنترلر می‌توانید مدارهای کاربردی متفاوتی را در سطح مقدماتی طراحی کنید و برد مدار چاپی آن را بسازید. برای تولید مدارهای پیشرفته، احتیاج به سخت‌افزارهای جانبی بیشتری در کنار میکروکنترلر است که درمقاطع تحصیلی بالاتر با کسب دانش بیشتر، توانایی و مهارت لازم را برای طراحی و اجرای پروژه‌های الکترونیکی پیچیده به دست می‌آورید.

استاندارد عملکرد

برنامه‌ریزی میکروکنترلر و استفاده از آن برای بستن مدار ساده و راه‌اندازی مدار با رعایت قوانین و استاندارد حاکم

۱-۷ مواد، ابزار و تجهیزات مورد نیاز

رایانه - نرم افزارهای مورد نیاز - بردبرد - میکروکنترلر ۸ ATmega - دستگاه پروگرامر (Programmer) - دیود LED قرمز و سبز - مقاومت $560\ \Omega$ - $470\ \Omega$ - $220\ \Omega$ - $3/3\ K\Omega$ ، $\frac{1}{4}$ وات - ترانزیستور BC ۱۴۰، BD ۱۳۷ - سیم های بردبرد - میز آزمایشگاهی الکترونیک با تجهیزات استاندارد - LCD 2×16 - هفت قطعه ای ۷ Seg (سون سگمنت) کاند مشترک - موتور DC کوچک - کلید شستی کوچک (SPST Push Button) - شارژر موبایل ۵ ولتی

۲-۷ انتخاب میکروکنترلر

چگونگی انتخاب میکروکنترلر مناسب برای اجرای پروژه های ساده الکترونیکی، در واحد یادگیری ۶ شرح داده شده است.

به منظور عملی شدن کنترل مدارهای الکترونیکی با میکروکنترلر، لازم است که دستورهای اجرایی را به وسیله نرم افزار Codevision AVR برنامه ریزی کنیم. پس از برنامه ریزی باید مدار با نرم افزار پروتئوس شبیه سازی شود. در صورت صحیح بودن برنامه و عملکرد درست مدار، برنامه را در میکروکنترلر به وسیله دستگاه پروگرامر (Programmer) بارگذاری می کنیم. بارگذاری برنامه در میکروکنترلر را اصطلاحاً پروگرام (Program) کردن برنامه می گویند. در این مبحث با اجرای یک کار عملی به صورت گام به گام با شیوه برنامه ریزی میکروکنترلر آشنا خواهید شد؛ سپس میکروکنترلر را به همراه مدار جانبی آن روی بردبرد نصب می کنید و مورد آزمایش قرار می دهید. در ادامه کار می توانید برد فیبر مدار چاپی آن را بسازید. برای رسیدن به این هدف و ایجاد انگیزه، ضرورت دارد مداری را انتخاب کنید که کاربردی باشد و عملاً بتوانید در زندگی روزمره خود استفاده کنید.

فیلم اجرای مدار چشمک زن با نرم افزار CodeVision را ببینید.

فیلم



کار عملی ۱



بستن مدار چشمک زن با دو LED به وسیله میکروکنترلر و برنامه نویسی آن با زبان C

هدف: کسب مهارت در برنامه ریزی، شبیه سازی و پروگرام کردن میکروکنترلر ۸ ATmega

مواد، ابزار و تجهیزات: لوازم التحریر - دستورهای زبان C - برگه اطلاعات میکروکنترلر ۸ ATmega - دستگاه پروگرامر (Programmer) - رایانه - نرم افزارهای مورد نیاز - بردبرد - میکروکنترلر ۸ ATmega - دیود LED قرمز و سبز هر کدام یک عدد - مقاومت $470\ \Omega$ ، $\frac{1}{4}$ وات دو عدد - سیم های بردبرد - میز آزمایشگاهی الکترونیک با تجهیزات استاندارد

مراحل اجرای کار:

۱ در شکل ۷-۱ مدار چشمک زن با دو LED نشان داده شده است. در این مدار زمان روشن و خاموش شدن، به وسیله میکروکنترلر تعیین می شود. برای این منظور بیت صفر و بیت ۱ از پورت B (پایه ۱۴ و ۱۵) میکروکنترلر را به صورت خروجی تعریف می کنیم. مقاومت های ۴۷۰ اهمی برای بایاس کردن دیودها به پایه های میکروکنترلر اتصال داده شده است.

سؤال

پالس ساعت (Clock pulse) اسیلاتور میکروکنترلر در مدار شکل ۷-۱، چگونه تولید شده است؟

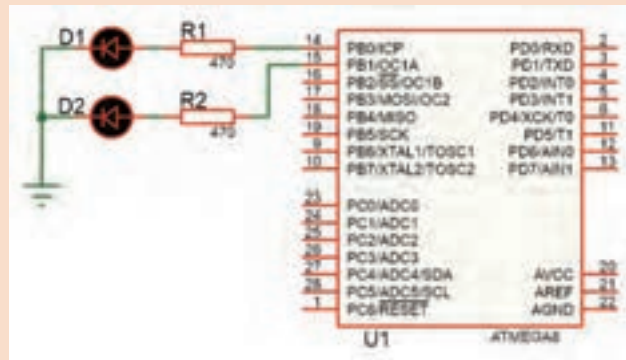
```
#include <delay.h>
void main(void)
while (1)
{
    // Place your code here

    PORTB.0=1;
    PORTB.1=0;

    delay_ms(1000);

    PORTB.0=0;
    PORTB.1=1;

    delay_ms(1000);
}
```



شکل ۷-۱ مدار چشمک‌زن با دو LED



شکل ۷-۲ برنامه مدار چشمک‌زن با زبان C

نرم‌افزار کدویژن (Codevision) را فعال کنید و با زبان برنامه‌نویسی C برنامه مدار چشمک‌زن را بنویسید.

نکته

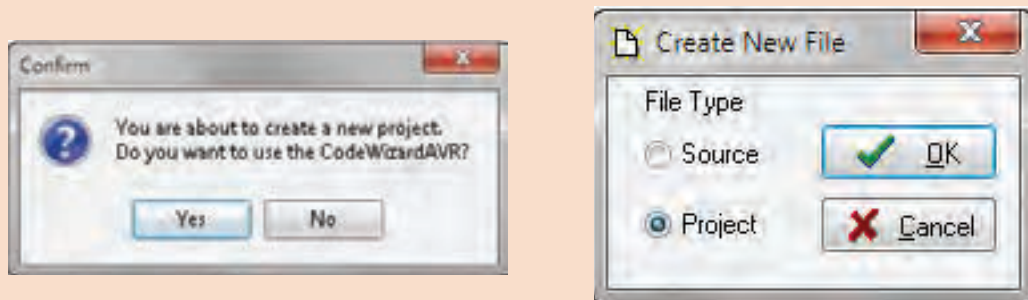
برای برنامه C در نرم‌افزار کدویژن از روش ایجاد پروژه با (Wizard) استفاده کنید.

برنامه مدار چشمک‌زن به زبان C را در شکل ۷-۲ مشاهده می‌کنید. با توجه به آموخته‌های خود دستورهای اجرای برنامه را شرح دهید.

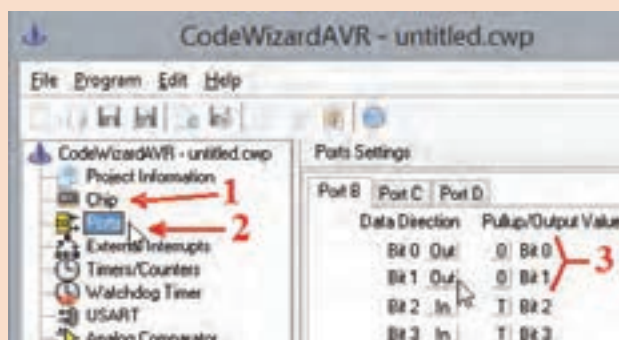
در برنامه Codevision از منوی file گزینه New را انتخاب کنید تا پنجره شکل ۷-۳ باز شود، سپس گزینه project را انتخاب کنید و دکمه ok را فعال کنید. برای استفاده از wizard روی گزینه yes شکل ۷-۴ کلیک کنید تا پنجره code wizard باز شود.

با انتخاب code wizard، پنجره تنظیمات ویزارد شکل ۷-۵ باز می‌شود.

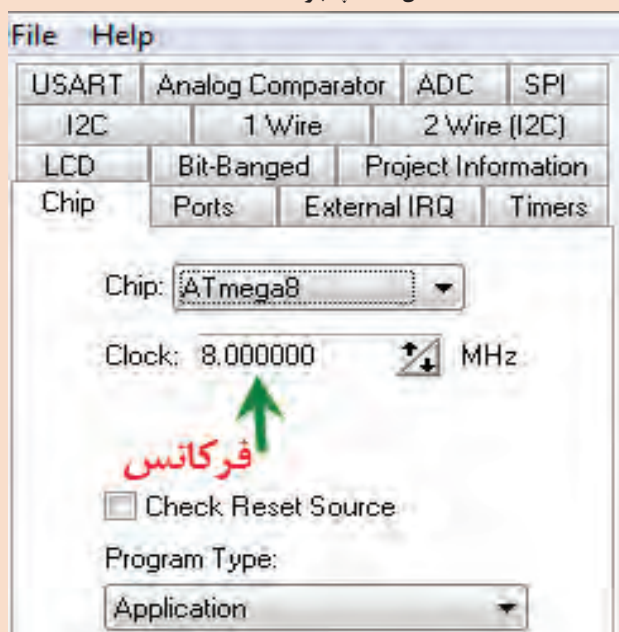
در مرحله اول در زبانه chip نوع میکروکنترلر ATmega ۸ را انتخاب کنید شکل الف ۷-۶.



شکل ۷-۳ پنجره انتخاب پروژه جدید در نرم افزار
پس از انتخاب ATmega ۸ پنجره بسته می شود. طبق شکل ب ۷-۶. مقدار فرکانس پالس ساعت (Clock Frequency) را روی ۸MHz بگذارید.

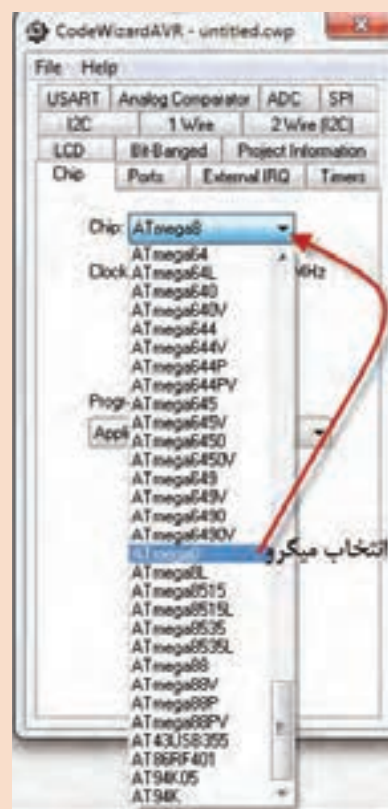


شکل ۷-۵ پنجره code wizard



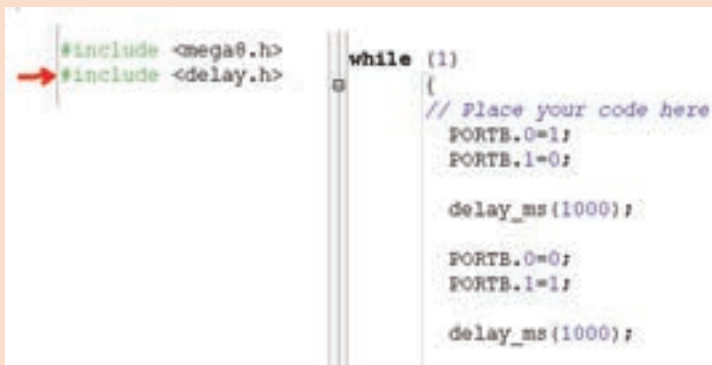
ب) تنظیم فرکانس میکروکنترلر

شکل ۷-۶ انتخاب میکروکنترلر از زبانه Chip



الف) انتخاب میکروکنترلر

در مرحله دوم با انتخاب زبانه Port در قسمت Port Setting، دو پایه‌ای را که می‌خواهید به‌عنوان خروجی استفاده کنید، بر روی حالت out بگذارید. شکل ۷-۵ در کادر code wizard با شماره ۳، قراردادن بیت صفر و یک از پورت B را در حالت خروجی نشان می‌دهد. بعد از اجرای تنظیمات از منوی program گزینه Generate، Save & Exit را انتخاب کنید، تا کد برنامه ایجاد شود. در کادرهای باز شده نام پروژه و نام برنامه را تایپ کنید. برای پروژه و برنامه از یک نام مشترک استفاده کنید.



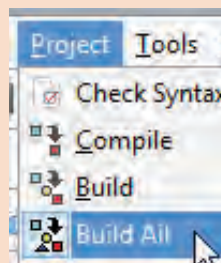
شکل ۷-۷ برنامه مدار چشمک‌زن code wizard

همان‌طور که در شکل ۷-۷ مشاهده می‌کنید قسمتی از متن برنامه توسط ویزارد (Wizard) نوشته شده است. در ادامه کار باید، فایل هدر <delay.h> را در ابتدای برنامه و زیر خط معرفی میکروکنترلر اضافه کنید. سپس در حلقه (۱) while متن برنامه را بنویسید.



شکل ۷-۹ نتیجه کار مترجم (Compiler) زبان C

بعد از کد نویسی برای بررسی خطاها و کامپایل کردن برنامه (ترجمه و تفسیر کردن) در شکل ۷-۸ از منوی project گزینه Build all را انتخاب کنید، در این حالت طبق شکل ۷-۹ پنجره اطلاعات پروژه باز می‌شود. در پنجره اطلاعات شکل ۷-۹، گزارشی از نتیجه Compiler و تعداد خطاهای برنامه نشان داده می‌شود. در صورتی که در برنامه خطایی وجود نداشته باشد روی گزینه OK کلیک کنید، تا برنامه ذخیره (SAVE) شود. در این حالت فایلی با پسوند hex برای برنامه‌ریزی کردن میکروکنترلر ایجاد می‌شود. ☒ فایل hex در پوشه Exe در محل ذخیره برنامه پروژه قرار می‌گیرد. ☒ فایل‌های دیگری جهت شبیه‌سازی و موارد دیگر نیز تولید می‌شود.



شکل ۷-۸ منوی Project و دستور ساختن Build All

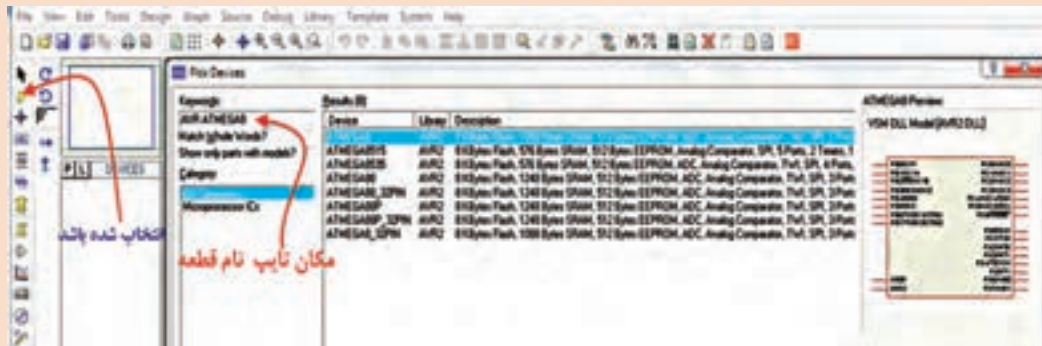
۸ نرم افزار Proteus را اجرا کنید.

در گوشه سمت چپ روی حالت (Component Mode) کلیک کنید، تا دسترسی شما به قسمت کتابخانه‌های نرم افزار امکان پذیر شود. برای این کار در قسمت DEVICES بر روی آیکن (Pick Devices) کلیک کنید شکل ۷-۱۰.



شکل ۷-۱۰ بخش ورود به کتابخانه

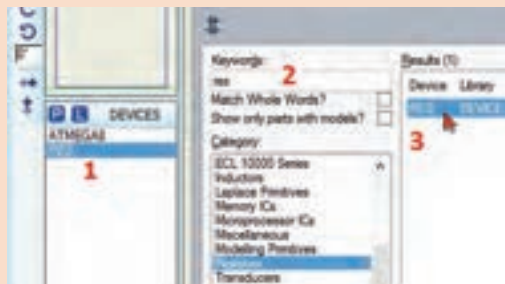
طبق شکل ۷-۱۱ در کادر باز شده Pick Devices، در قسمت Keywords می‌توانید نام قطعه یا کتابخانه مورد نظر را جستجو کنید. کلمه AVR را تایپ کنید، در این قسمت فهرست میکروکنترلرهای خانواده AVR به نمایش در می‌آید، از این فهرست میکروکنترلر ۸ ATmega را انتخاب کنید. با دو بار کلیک کردن بر روی قطعه مورد نظر، قطعه وارد قسمت DEVICES می‌شود.



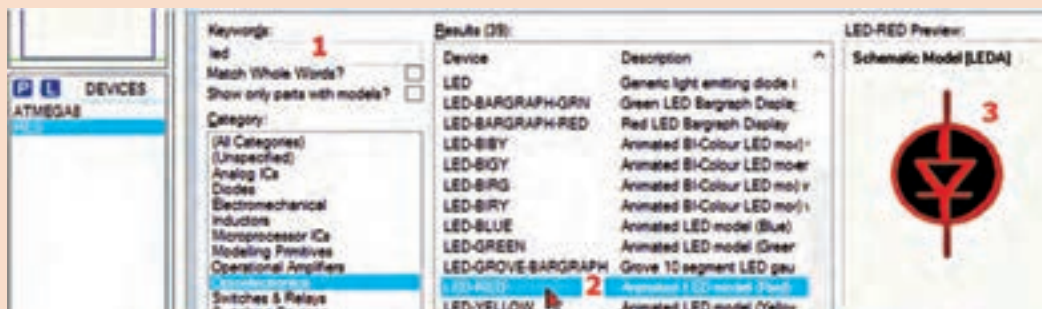
شکل ۷-۱۱

طبق شکل ۷-۱۲ عبارت res را در مکان Keywords بنویسید و مقاومت مدار را انتخاب کنید. با دوبار کلیک کردن روی Resistors، مقاومت وارد قسمت DEVICES می‌شود.

برای دیودهای نورانی کلمه LED را بنویسید. طبق شکل ۷-۱۳ در قسمت نتیجه جستجو، LED با رنگ‌های سبز و قرمز را انتخاب کنید.

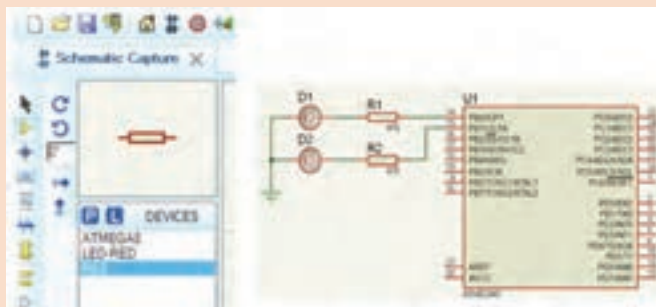


شکل ۷-۱۲ نحوه انتخاب مقاومت



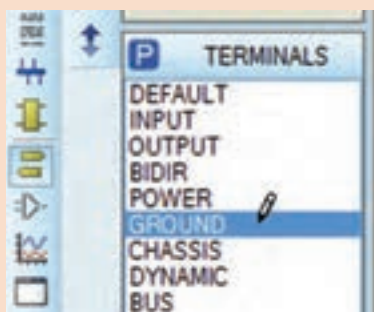
شکل ۷-۱۳ نحوه انتخاب LED

قطعات را از جعبه DEVICES به محیط کار پروتئوس بیاورید و طبق شکل ۷-۱۴ پایه‌های میکروکنترلر را به LEDها سیم‌کشی کنید.




شکل ۷-۱۴ انتقال قطعه و رسم نقشه مدار برای شبیه‌سازی

برای اتصال Vcc و GND در نرم‌افزار پروتئوس، طبق شکل ۷-۱۵ در آیکن Terminals Mode گزینه‌های POWER و GROUND را انتخاب کنید.



بعد از اینکه سیم‌کشی مدار تمام شد، باید برنامه نوشته شده را در میکروکنترلر بارگذاری کنیم، برای این کار بر روی نماد فنی میکروکنترلر در نقشه دوبار کلیک کنید، تا کادر شکل ۷-۱۶ باز شود.

با کلیک بر روی  در قسمت Program File، فایل (hex یا cof) را از مسیر ذخیره شده برنامه میکروکنترلر را انتخاب کنید.

شکل ۷-۱۵ انتخاب پایه‌های Power، GND

فرکانس کار میکروکنترلر را مطابق انتخاب اولیه در محیط برنامه‌نویسی کدوین (MHz Int RC) تغییر دهید و کلید ok را بزنید، میکروکنترلر در نرم‌افزار آماده اجرا می‌شود.



شکل ۷-۱۶ پنجره ویرایش و تنظیم میکروکنترلر برای اجرا در پروتئوس

برای اجرای برنامه نوشته شده در گوشه پایین سمت چپ نرم افزار پروتئوس بر روی start (دکمه PLAY) کلیک کنید تا برنامه اجرا شود شکل ۱۷-۷.

شکل ۱۷-۷ میکروکنترلر در حال اجرا

✓ اگر زمان‌های خاموش و روشن شدن LEDها زیاد است، در برنامه چشمک‌زن، زمان تأخیر را به $delay_ms(500)$ تغییر دهید.

✓ به مقدار فرکانس کار میکروکنترلر توجه کنید.

توجه



۹ برنامه را به وسیله پروگرامر موجود در هنرستان با راهنمایی و نظارت مربی کارگاه به میکروکنترلر انتقال دهید.

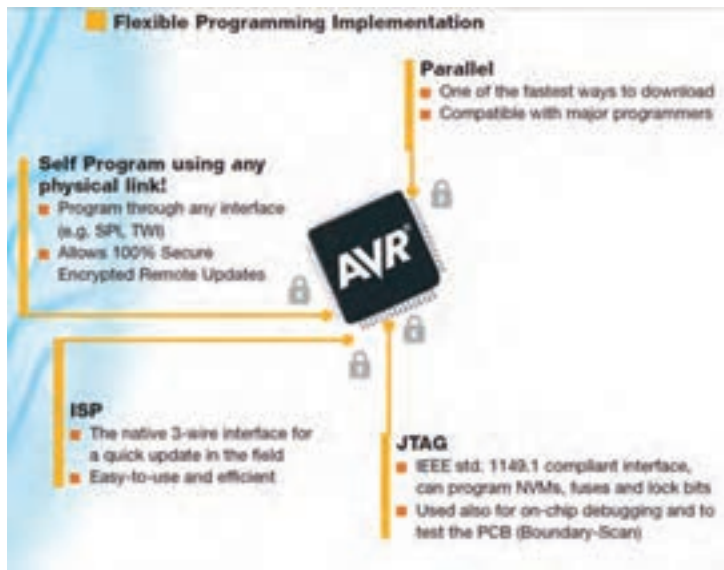
۱۰ میکروکنترلر و قطعات جانبی آن را روی بردبرد ببندید. اتصال تغذیه مدار را از منبع تغذیه آزمایشگاهی ۵ ولت ثابت یا یک شارژر موبایل ۵ ولتی برقرار کنید.

۱۱ عملکرد مدار چشمک‌زن را مشاهده کنید. در صورتی که مدار فعال نشد، اتصال پایه‌های میکروکنترلر و جهت قرارگیری آند و کاتد دیودها را کنترل کنید. در غیر این صورت به کمک مربی کارگاه، مدار را بازرینی و عیب‌یابی کنید. پس از رفع عیب، دوباره مدار راه‌اندازی کنید.

۱۲ نتایج حاصل از راه‌اندازی مدار چشمک‌زن را تحلیل کنید و به‌طور خلاصه بنویسید.

۷-۳ پروگرام کردن میکروکنترلر

اغلب میکروکنترلرهای خانواده AVR به چهار روش قابل برنامه‌ریزی یا پروگرام شدن هستند. در شکل ۷-۱۸ این چهار روش انتقال برنامه به داخل میکروکنترلر را نشان می‌دهد.



شکل ۷-۱۸ روش‌های پروگرام کردن برخی از میکروکنترلرهای خانواده AVR

متن تصویر ۷-۱۸ را به زبان فارسی ترجمه کنید.

ترجمه کنید



شرکت‌های سازنده میکروکنترلر، پروگرامرهای مختلفی برای AVR ساخته‌اند، یکی از رایج‌ترین و ساده‌ترین نوع آنها مدلی است که ورودی آن به درگاه پرینتر (پورت پارالل) رایانه وصل می‌شود و خروجی آن به روش ISP (واسطه سه سیمه)، میکرو را پروگرام می‌کند. چنانچه پروگرامرها با استاندارد شرکت ATMEL ساخته شوند، با نرم‌افزار Codvision شناخته می‌شوند. بعضی از پروگرامرها نرم‌افزارهای مخصوص خود را دارند. در هر صورت اگر نرم‌افزارها متفاوت باشند قابلیت سازگاری با میکروکنترلرهای مختلف را دارد. معمولاً در این نرم‌افزارها علائم و کلیدهای مشابه وجود دارد که امکان پروگرام کردن میکروکنترلرها را فراهم می‌کند. در شکل ۷-۱۹ دو نمونه از پروگرامر میکروکنترلر نشان داده شده است.

با مراجعه به رسانه‌های مختلف، در رابطه با انواع پروگرامرهای میکروکنترلر تحقیق کنید و نتیجه تحقیق خود را ارائه دهید.

تحقیق کنید



تفاوت‌های بین پورت پرینتر قدیمی در کامپیوتر و پورت USB را شرح دهید.

سوال و جست و جو





شکل ۱۹-۷ پروگرامر میکروکنترلر

■ فیوزبیت‌ها (Fuse Bits)

فیوزبیت‌ها، بخش رجیستری (ثبت Register) حافظهٔ فلش هستند، که با تغییر آنها (صفر یا یک کردن) امکاناتی را در اختیار کاربر قرار می‌دهند. فیوزبیت‌های میکروکنترلرهای AVR در هر پروژه برای شروع به کار نیاز به تنظیم اولیه دارد.

فیوزبیت‌ها برای تنظیم‌های خاصی استفاده می‌شوند و با پاک کردن میکروکنترلر از بین نمی‌روند. لذا تغییر در آنها فقط از طریق پروگرامر امکان‌پذیر است، همچنین برای تنظیم آنها نیاز به برنامه‌نویسی خاص نداریم. میکروکنترلرها بسته به نوع قابلیت‌هایی که دارند دارای فیوزبیت‌های متفاوتی هستند. برخی از فیوزبیت‌ها عبارت‌اند از:

- ☑ فیوز CKSEL برای انتخاب منبع پالس ساعت clock pulse میکروکنترلر به صورت داخلی یا به صورت خارجی با کریستال پیزو الکتریک به کار می‌رود.
- ☑ اگر بخواهیم پس از پاک کردن برنامه میکروکنترلر، حافظه EEPROM آن را داشته باشیم یا حذف کنیم از فیوزبیت EESAVE استفاده می‌کنیم.

در صورت برنامه‌ریزی فیوزبیت BOOTRST، آدرس Reset را به ابتدای حافظه Boot انتقال می‌دهد. در این وضعیت برای راه‌اندازی دوباره میکروکنترلر نیاز به خاموش و روشن کردن تغذیه DC نیست. این نوع راه‌اندازی (Boot) معمولاً روال خودکاری از مجموعه‌ای دستورهای است که با روشن کردن

یا بازنشانی (Reset) میکروکنترلر شروع می‌شود. همچنین، عملکرد آن مشابه عملکرد کلید فشاری Reset در رایانه است.

اگر به فیوزبیت‌ها ارزش صفر "۰" بدهیم به معنای بیت برنامه‌ریزی شده (PROGRAMMED) و چنانچه ارزش "۱" بدهیم به معنای بیت برنامه‌ریزی نشده (UNPROGRAMMED) است.

با مراجعه به رسانه‌های مختلف، اطلاعات مرتبط به فیوزبیت‌های خانواده میکروکنترلرهای AVR را بیابید و آنها را استخراج کنید و نتیجه تحقیق خود را در قالب یک گزارش ارائه دهید.

تحقیق کنید



پروگرام کردن میکروکنترلر به وسیله نرم‌افزار

هدف: کسب مهارت در CodeVision برای پروگرام کردن میکروکنترلر ۸ ATmega
مواد، ابزار و تجهیزات: برگه اطلاعات میکروکنترلر ۸ ATmega - دستگاه پروگرامر (Programmer) -
رایانه - نرم‌افزارهای مورد نیاز - میکروکنترلر ۸ ATmega

کار عملی ۲



با مراجعه به رسانه‌های مختلف نحوه ورود به بخش پروگرامر و انتخاب نوع و تنظیم سرعت آن در نرم‌افزار Codevision را پیدا کنید.

جست و جو کنید



فیلم برنامه‌ریزی کردن میکروکنترلر با پروگرامر را ببینید.

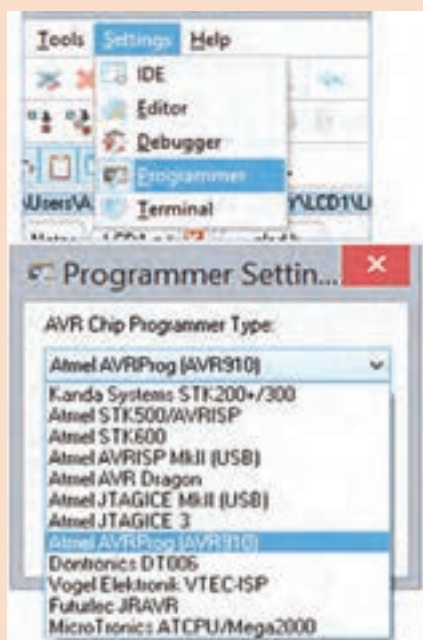
فیلم



مراحل اجرای کار:

۱ طبق شکل ۷-۲۰ به پنجره Settings از منوی اصلی در نرم‌افزار Codevision وارد شوید، در داخل زبانه Programmer و از فهرست آن، نوع پروگرامر موجود در کارگاه یا پروگرامری را که برای خودتان خریداری کرده‌اید انتخاب کنید. در شکل ۷-۲۰ نوع پروگرامر مدل Atmel AVRProg (AVR۹۱۰) انتخاب شده است.

۲ سرعت انتقال و درگاه (Port) انتقالی که پروگرامر به آن وصل شده است را مطابق شکل ۷-۲۱ تنظیم کنید.



شکل ۷-۲۰ مسیر انتخاب پروگرامر Codevision

اگر پورتهی که پروگرامر به آن وصل است را نمی‌شناسید به قسمت Device Manager در Control Panel مراجعه کنید، در قسمت Port&LPT نام پروگرامر و پورت Com آن مشخص شده است.

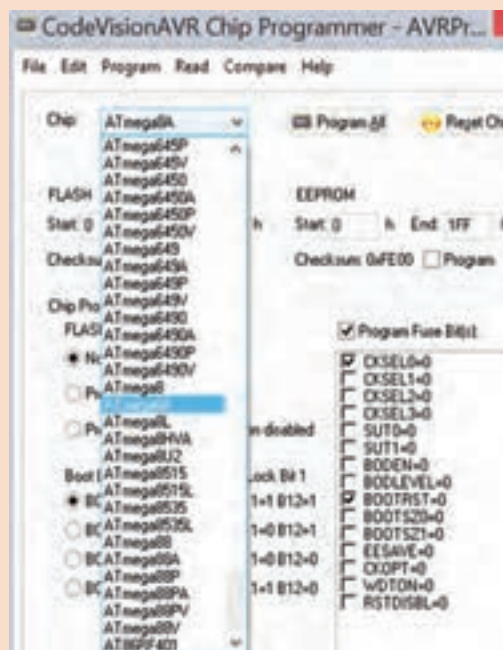
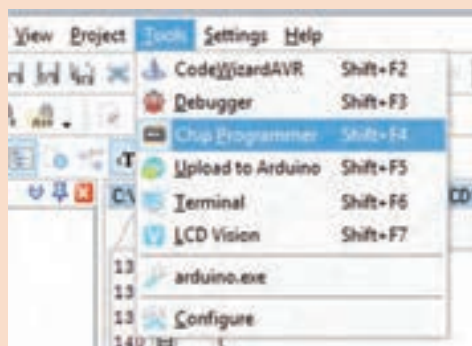
توجه



۳ از منوی Tools شکل ۷-۲۲، گزینه Chip Programmer را انتخاب کنید. تیک گزینه Program Fuse Bit را بزنید تا فیوزبیت‌های میکروکنترلر برنامه‌ریزی شوند.



شکل ۷-۲۱ تنظیم پورت و سرعت پروگرامر AVR ۹۱۰



شکل ۷-۲۲ تنظیمات پروگرامر

با مراجعه به رسانه‌های مختلف مشخص کنید برای غیر قابل کپی شدن برنامه که اصطلاحاً آن را قفل کردن (Lock) میکروکنترلر می‌گویند چه کاری باید انجام داد؟

جست و جو کنید



۴ میکروکنترلر را روی پروگرامر و در محل مناسب و درجهت، صحیح نصب کنید. شکل ۷-۲۳، شماره ۱ محل قرار گرفتن میکروکنترلر را در پروگرامر نشان می‌دهد.

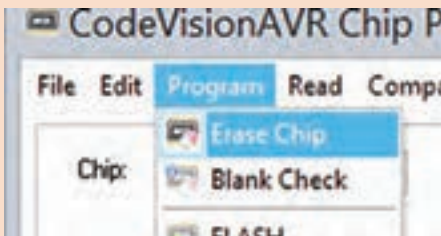
۵ از منوی File گزینه Load FLASH را انتخاب و فایل HEX ساخته شده با نرم‌افزار کدویژن (برنامه مدار پروژه چشمک‌زن) را بارگذاری کنید.



شکل ۷-۲۳ سوکت ZIP برای قراردادن میکرو

مثال

با فرض اینکه پروژه به نام Test یا فایل‌های دیگر مرتبط با آن مانند D:\STU در این آدرس قرار دارند، فایل HEX شما در آدرس D:\STU\Debug\Exe\Test. قرار دارد. HEX وجود دارد.



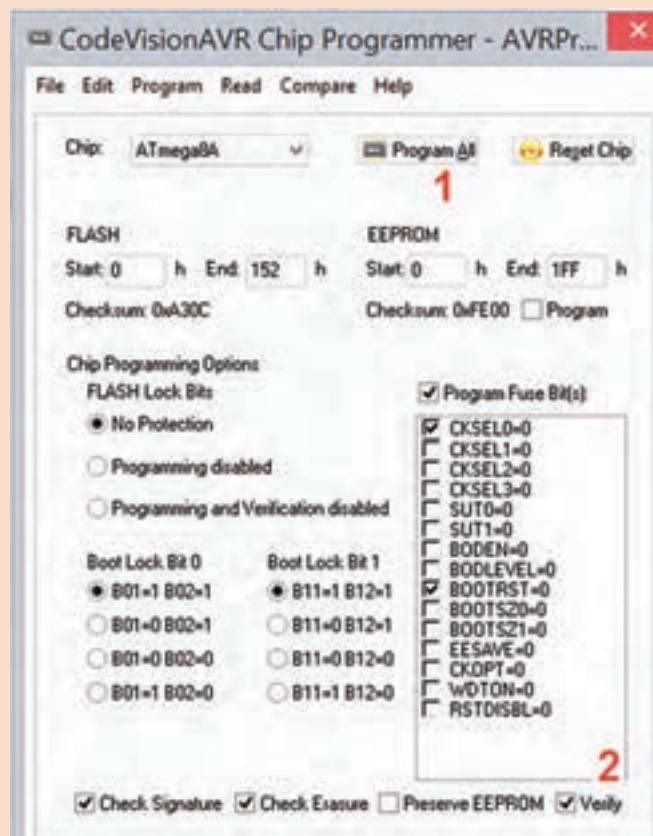
۶ طبق شکل ۷-۲۴ میکروکنترلر را پاک کنید و تنظیمات مدل کلاک و سرعت کلاک و قفل میکروکنترلر را انجام دهید. سپس با فعال کردن کلید Program All میکروکنترلر را برنامه‌ریزی کنید، طبق شکل ۷-۲۵، با زدن تیک ✓ روی گزینه Verify از اجرای برنامه در نرم‌افزار اطمینان حاصل می‌شود.

۷ میکروکنترلر و قطعات جانبی آن را روی برد برد بندید. تغذیه مدار را از منبع تغذیه آزمایشگاهی ۵ ولت ثابت یا یک شارژر موبایل ۵ ولتی تأمین کنید.

۸ مدار را راه‌اندازی کنید و عملکرد مدار چشمک‌زن را مشاهده کنید.

۹ مدار پروژه چشمک‌زن را برای ارزشیابی به مربی کارگاه ارائه دهید.

۱۰ گزارش مختصری از این فعالیت بنویسید.



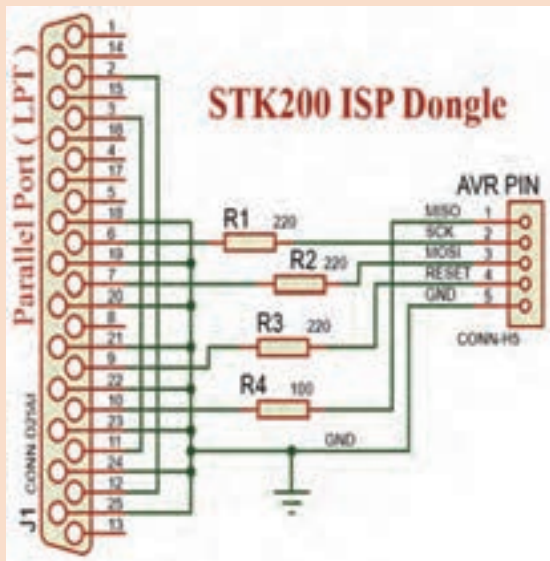
شکل ۷-۲۵ تنظیم نهایی پروگرامر



ساخت پروگرامر میکروکنترلر

هدف: کسب مهارت در طراحی و مونتاژ یک نمونه پروگرامر میکروکنترلر
مواد، ابزار و تجهیزات: برگه اطلاعات میکروکنترلر ATmega ۸ - رایانه - نرم افزارهای مورد نیاز - میکروکنترلر ATmega ۸ - مقاومت های $220\ \Omega$ - $220\ \Omega$ - $100\ \Omega$ - وات سه عدد و مقاومت $100\ \Omega$ - وات یک عدد - کانکتور LPT یک عدد - کانکتور AVR یک عدد - پین هدر ۵ پین یک عدد

مراحل اجرای کار



شکل ۲۶-۷ یک نمونه پروگرامر

- ۱ همان طور که در شکل ۲۶-۷ مشاهده می کنید این پروگرامر دارای چهار عدد مقاومت است و سیم کشی آن با استفاده از پورت پرینتر ساخته شده و با نرم افزار کدویژن شناسایی می شود. مدل آن STK۲۰۰ است و برنامه ریزی آن با روش ISP انجام می شود. پایه های MISO، SCK، MOSI، RESET، GND به میکروکنترلر وصل می شود. با راهنمایی مربی کارگاه، برد فیبر مدار چاپی را برای این پروگرامر طراحی کنید.
- ۲ قطعات را روی فیبر مونتاژ کنید و به رایانه متصل کنید.
- ۳ برنامه چشمک زن را در میکروکنترلر برنامه ریزی کنید.

نکته



از آنجا که رایانه های امروزی فاقد پورت (Parallel Port LPT) هستند، لازم است پروگرامری ساخته شود که بتواند به ورودی USB کامپیوتر وصل شود. در این حالت در صورت داشتن مبدل LPT به USB، این کار با همین پروگرامر امکان پذیر می شود. در غیر این صورت باید از مدار دیگری که دارای IC درایور و قطعات جانبی دیگری است، استفاده شود.

پژوهش



با مراجعه به رسانه های مختلف نمونه هایی از پروگرامرهای جدید را بیابید و در قالب یک گزارش ارائه کنید.

۴ مدار را راه اندازی کنید و آن را برای ارزشیابی به مربی کارگاه نشان دهید.

نکته



پروگرامر میکروکنترلر را اصطلاحاً دانگل (Dongle) می نامند. دانگل به معنی وسیله کوچکی است که به وسیله آن می توانیم با کامپیوتر ارتباط برقرار کنیم. دانگل در واقع یک واسط است.



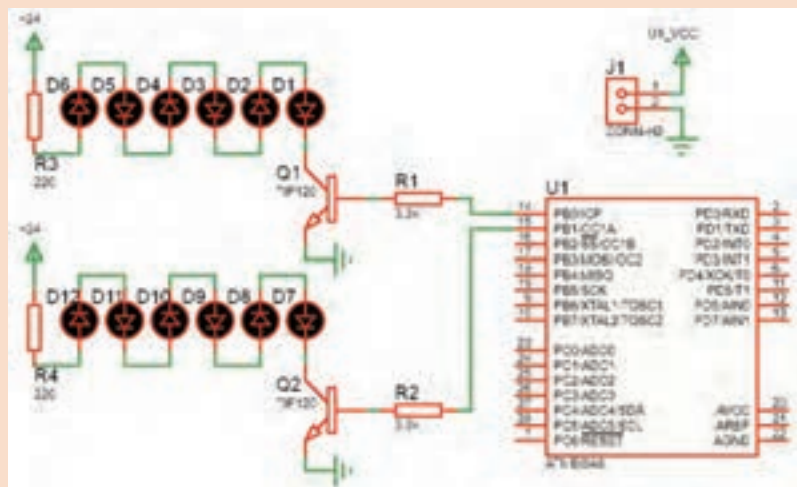
بستن مدار چشمک‌زن دو کاناله به وسیله میکروکنترلر

هدف: کسب مهارت در برنامه‌ریزی، شبیه‌سازی، پروگرام کردن میکروکنترلر ATmega ۸

مواد، ابزار و تجهیزات: لوازم التحریر- دستوره‌ای زبان C- برگه اطلاعات میکروکنترلر ATmega ۸ - دستگاه پروگرامر (Programmer)- رایانه - نرم‌افزارهای مورد نیاز- بردبرد- میکروکنترلر ATmega ۸ - دیودهای LED قرمز و سبز هر کدام ۶ عدد- مقاومت‌های $3/3\text{ K}\Omega$ ، $1/4$ وات و $220\ \Omega$ ، $1/4$ وات - سیم‌های بردبرد - ترانزیستور TIP۱۲۰ یا BD۱۳۷ دو عدد - میز آزمایشگاهی الکترونیک با تجهیزات استاندارد

مراحل اجرای کار

۱ مدار شکل ۲۷-۷ برای کنترل تعداد زیادی LED به صورت هم‌زمان به کار می‌رود. به این مدار، مدار چشمک‌زن دو کاناله نیز می‌گویند. ترانزیستورها به عنوان تقویت‌کننده جریان و راه‌انداز و برای افزایش جریان مورد نیاز دیودهای LED در مدار به کار رفته است. این مدار در تابلوهای LED با نور ثابت کاربرد دارد.



شکل ۲۷-۷ کنترل چند LED

مقدار مقاومت سری با LED ها با توجه به مقدار ولتاژ کار LED، جریان مصرف LED و ولتاژ تغذیه تعیین می‌شود. با توجه به نوع LED موجود، مقدار مقاومت را با کمک مربی خود به دست آورید.

نکته



۲ با استفاده از جدول ۷-۱ و به وسیله زبان برنامه‌نویسی C و نرم‌افزار کد ویرن برنامه چشمک‌زن دو کاناله را بنویسید.

۳ مدار شکل ۲۷-۷ را در نرم‌افزار پروتئوس Proteus شبیه‌سازی و اجرا کنید.

۴ برنامه را به وسیله پروگرامر ساخته شده یا با پروگرامر موجود در کارگاه به میکروکنترلر انتقال دهید.

۵ مدار را روی بردبرد ببندید. برای تغذیه از منبع آزمایشگاهی ۵ ولت ثابت، یا یک شارژر موبایل ۵ ولتی استفاده کنید. تغذیه LED ها را نیز از ولتاژ ۲۴ ولت منبع تغذیه آزمایشگاهی تأمین کنید.

۶ با تغییر در برنامه، زمان‌های خاموش و روشن شدن چشمک‌زن را تغییر دهید.

جدول ۷-۱ برنامه به زبان C

```
قبل از void main(void) خط زیر
#include <delay.h> اضافه شود و
در زیر (۱) برنامه نوشته شود
while (۱)
{
    // Place your code here
    PORTB.۰=۱;
    PORTB.۱=۱;
    delay_ms(۱۰۰۰);
    PORTB.۰=۰;
    PORTB.۱=۰;
    delay_ms(۱۰۰۰);
}
```

- ✓ اگر زمان را به کمتر از ۱۰ میلی‌ثانیه کاهش دهیم به نظر شما چه اتفاقی می افتد؟
- ✓ اگر مقاومت های R_p و R_p را به سه برابر افزایش دهیم، چه تغییری در عملکرد مدار رخ می‌دهد؟
- ✓ تغییرات خواسته شده را انجام داده و نتایج و علت را در کارگاه به بحث بگذارید.

پژوهش



۷ اگر بخواهیم بدون تغییرات در سخت افزار مدار، نور LEDها را کاهش دهیم، چه تغییراتی باید در برنامه ایجاد کنیم؟

۸ اگر تعداد کانالها را بخواهیم افزایش دهیم (چهار کاناله)، تغییراتی را که در نقشه سخت افزار باید ایجاد کنیم رسم کنید و تغییری که در برنامه لازم است را بنویسید.

۹ نقشه جدید را پس از رسم و تهیه برنامه از منابع مختلف به زبان C، برای ارزشیابی به مربی کارگاه بدهید.

پروژه مدار چشمک زن ۸ کاناله را با طراحی مدار چاپی بسازید. (فقط با ولتاژ ۲۴ ولت)

فعالیت



۷-۴ مدارهای کاربردی با میکروکنترلر

وقتی که قرار است یک پروژه با میکروکنترلرهای AVR انجام داده شود، بعد از مشخص شدن هدف پروژه و مقرون به صرفه بودن آن باید مراحل طراحی سخت افزار، طراحی نرم افزار (الگوریتم یا فلوچارت برنامه ریزی)، برنامه نویسی، شبیه سازی و پیاده سازی (پروگرام کردن میکروکنترلر و بستن مدار مورد نظر روی برد) انجام شود.

میکروکنترلرها توانایی های شگفت آور ندارند و قدرت پردازش آنها محدود است. این تراشه ها برای کارهای تخصصی مانند کنترل اسباب بازی ها، دستگاه های پزشکی، ماشین های اداری، موتور ها، دستگاه های الکترونیک خانگی و کنترل کننده های از راه دور به کار می روند. برای آشنایی بیشتر با عملکرد آنها در این بخش کارهای عملی ساده با میکروکنترلر معرفی شده است. با اجرای متمرکز و دقیق این فعالیت ها، شما می توانید توانایی و مهارت انجام کار با مدارهای کاربردی میکروکنترلر را کسب کنید. در اجرای کارهای عملی سعی شده است، دستورهای برنامه نویسی C متناسب با هر کاربرد میکروکنترلر شرح داده شود.

■ ارسال اطلاعات به خروجی

در صورتی که بخواهیم همه پایه های یک پورت را روشن و یا خاموش کنیم چه باید کرد؟ آیا باید به صورت جداگانه تک تک پایه ها را روشن و یا خاموش کنیم؟ درباره این موضوع جست و جو کنید و نتیجه را از طریق بارش فکری در کارگاه به بحث بگذارید.

بارش فکری



شکل ۷-۲۸ ارسال عدد باینری به پورت



شکل ۷-۲۹ ارسال عدد هگزادسیمال به یک پورت



شکل ۷-۳۰ ارسال عدد دسیمال به یک پورت

اگر بخواهیم همه پایه های یک پورت را روشن و یا خاموش کنیم باید مقدار داده (Data) را به هر پورت وارد کنیم. هنگامی که یک پورت (هشت پایه) به عنوان خروجی تعیین می شود می توان اطلاعات را به صورت های مختلفی به پورت مورد نظر ارسال کرد. **✓ ارسال اطلاعات به صورت باینری:** در این

حالت با توجه به اینکه هر پورت دارای هشت پایه است، می توان ۸ عدد صفر یا ۸ عدد یک نوشته شود. که در واقع نشانگر یک بایت می باشد. در این بایت کم ارزش ترین بیت مربوط به پایه صفر از پورت است، (مانند PC یا PB) و با ارزش ترین بیت مربوط به پایه ۷ (مانند PC یا PB) از پورت مورد نظر است. به الگوی عدد ارسال شده به پورت B در شکل ۷-۲۸ توجه کنید، این روش ارسال به صورت باینری است. محدوده عدد ارسالی به صورت :

۰b۱۱۱۱۱۱۱۱ ~ ۰b۰۰۰۰۰۰۰۰ می باشد.

✓ **ارسال اطلاعات به صورت هگزادسیمال:** در این حالت عدد مورد نظر به مبنای هگزادسیمال یا شانزده تایی تبدیل می‌شود. هر ۸ بیت باینری معادل ۲ عدد در مبنای هگزادسیمال است. هر کدام از این ۲ عدد، اطلاعات ۴ پایه از میکروکنترلر را شامل می‌شود. به الگوی شکل ۷-۲۹ دقت کنید. محدوده عدد ارسالی: $0x00 \sim 0xFF$ است.

✓ **ارسال اطلاعات به صورت دسیمال:** در این حالت اطلاعات در مقابل پورت‌ها به صورت دسیمال (عدد طبیعی) نوشته می‌شود. در شکل ۷-۳۰ محدوده عدد ارسالی می‌تواند بین صفر ۰ تا ۲۵۵ باشد. روش ارسال به صورت ده‌دهی (دسیمال) کمتر مورد استفاده قرار می‌گیرد و مترجم زبان برنامه‌نویسی این عدد را به مبنای باینری تبدیل می‌کند.

کار عملی ۵



مدار چشمک‌زن با دو LED با دکمه فشاری (شستی) شروع به کار (Start)

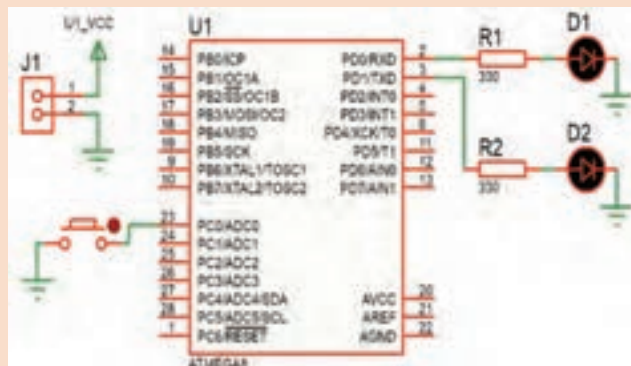
هدف: کسب مهارت در برنامه‌نویسی، شبیه‌سازی، پروگرام کردن میکروکنترلر ۸ Atmega و بستن مدار سخت افزاری

مواد، ابزار و تجهیزات: لوازم التحریر- دستوره‌های زبان C- برگه اطلاعات میکروکنترلر ۸ ATmega- دستگاه پروگرامر (Programmer)- رایانه- نرم‌افزارهای مورد نیاز- بردبرد- میکروکنترلر ۸ ATmega- میز آزمایشگاهی الکترونیک با تجهیزات استاندارد- دیود LED قرمز و سبز هر کدام یک عدد- مقاومت 330Ω ، $\frac{1}{4}$ وات دو عدد- سیم‌های بردبرد- کلید فشاری کوچک (SPST Push Button)

مراحل اجرای کار

۱ **مدار نشان داده شده در شکل ۷-۳۱ مشابه کار عملی شماره ۱ است، با این تفاوت که یک کلید برای تعیین حالت روشن یا خاموش یا تغییر وضعیت نمایش به کار رفته است.** مدار را در نرم‌افزار Proteus رسم و شبیه‌سازی کنید.

۲ **در نرم‌افزار کدویژن با توجه به خواسته‌های نقشه فنی، پایه‌های PD.۰، PD.۱ را در حالت خروجی و پایه PC.۰ را در حالت ورودی با وضعیت (P) Pullup قرار دهید، شکل ۷-۳۲.**



۳ **در حالت اول، جدول ۷-۲ برنامه به زبان C در نرم‌افزار کدویژن را طوری بنویسید که در مدار چشمک‌زن هم زمان با فشردن کلید فشاری، هر دو LED باهم شروع به چشمک‌زدن کنند.**

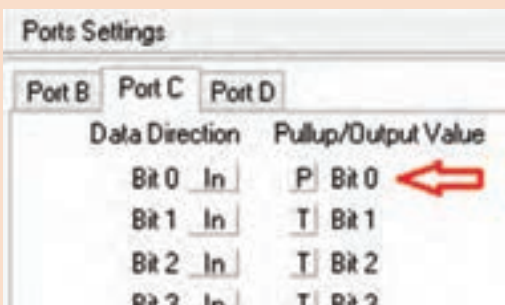
۴ **تمام مراحل شبیه‌سازی مدار را در نرم‌افزار پروتئوس انجام دهید.**

۵ **برنامه را توسط پروگرامر به میکروکنترلر انتقال دهید.**

۶ **مدار سخت‌افزاری چشمک‌زن میکروکنترلر را روی بردبرد ببندید.**

شکل ۷-۳۱ چشمک‌زن با کلید فشاری Start جهت کنترل

۷ چگونگی روشن شدن LED های خروجی مدار را با توجه به فشار دادن کلید فشاری به طور خلاصه توضیح دهید.



۸ در حالت دوم، برنامه را طبق جدول ۷-۲ به گونه ای تغییر دهید که با وصل کلید فشاری، LED متصل به PD.۰ و در صورت رها شدن کلید LED متصل به PD.۱ چشمک زن شود.

۹ تمام مراحل شبیه سازی مدار را در نرم افزار پروتئوس انجام دهید.

۱۰ برنامه را توسط پروگرامر به میکروکنترلر انتقال دهید.

شکل ۷-۳۲ تنظیم برای کلید فشاری

۱۱ مدار سخت افزاری چشمک زن میکروکنترلر را روی برد برد ببندید. جدول ۷-۲

برنامه به زبان C برای حالت ۱	برنامه به زبان C برای حالت ۲
<pre>while (1) { // Place your code here if(PINC.0==0){ PORTD=0B11; delay_ms(500); PORTD=0B00; delay_ms(500); } }</pre>	<pre>while (1) { // Place your code here if(PINC.0==0){ PORTD=0B01; delay_ms(500); PORTD=0B00; delay_ms(500); } else{ PORTD=0B10; delay_ms(500); PORTD=0B00; delay_ms(500); } }</pre>

۱۲ چگونگی روشن شدن LED های خروجی مدار را با توجه به حالت های مختلف کلید فشاری به طور خلاصه توضیح دهید.

۱۳ برنامه میکروکنترلر، در حالت اول را با حالت دوم مقایسه کنید. تفاوت های آنها را به اختصار شرح دهید.

۱۴ برای خواندن پایه PC.۰ از دستور PINC.۰ و برای مقایسه از علامت (==) استفاده شده است. چرا در برنامه کنترل میکروکنترلر عمل مقایسه پایه PC.۰ با عدد ۰ (if(PINC.۰==۰)) انجام می شود؟ توضیح دهید.

برای مدار شکل ۷-۳۱ دو روش دیگر برای چشمک زدن LED ها پیشنهاد دهید. مدار آنها را طراحی کنید. مدار سخت افزاری طرح های پیشنهادی را ببندید.

فعالیت



تحقیق کنید



فیلم



کار عملی ۶



با مراجعه به رسانه‌های مختلف، اطلاعات مرتبط به روش‌های کنترل سرعت موتور را تحقیق و استخراج کنید و نتیجه تحقیق خود را ارائه دهید.

فیلم روش کنترل دور موتور را ببینید.

کنترل موتور DC با کلید فشاری شروع به کار (Start)

هدف: کسب مهارت در برنامه‌نویسی، شبیه‌سازی، پروگرام کردن میکروکنترلر ۸ Atmega و بستن مدار سخت افزاری

مواد، ابزار و تجهیزات: لوازم التحریر- دستوره‌های زبان C- برگه اطلاعاتی میکروکنترلر ۸ Atmega- دستگاه پروگرامر (Programmer)- رایانه- نرم‌افزارهای مورد نیاز- بردبرد- میکروکنترلر ۸ Atmega- میز آزمایشگاهی الکترونیک با تجهیزات استاندارد- دیود ۱N۴۰۰۱ یک عدد- مقاومت $10K\Omega$ ، $\frac{1}{4}$ وات یک عدد- سیم‌های بردبرد- کلید فشاری کوچک (SPST Push Button)- ترانزیستور BD۱۳۷ یک عدد- موتور DC کوچک کم وات

مراحل اجرای کار:

✓ یکی از رایج‌ترین و پرکاربردترین موتورهای الکتریکی، موتورهای جریان مستقیم یا DC motor ها است. سرعت چرخش موتورهای DC به ولتاژ ورودی و قدرت این موتورها به جریان عبوری از آنها بستگی دارد. موتورهای DC به دلیل گشتاور بالا در سرعت کم کاربردهای بسیار زیادی دارند. از کاربردهای آنها می‌توان به ساخت روبات‌ها، شیشه بالابر خودروها، سشوارها، دریل‌های شارژی، انواع ماشین‌های کنترلی، لوازم ورزشی مانند تردمیل‌ها و اسباب بازی اشاره کرد.

۱ مدار شکل ۷-۳۳ یک مثال ساده از راه‌اندازی یک موتور DC است که از یک کلید فشاری برای تعیین حالت روشن و خاموش یا تغییر وضعیت دور موتور استفاده شده است.

۲ مدار شکل ۷-۳۳ را در نرم‌افزار Proteus رسم کنید.

۳ در نرم‌افزار کد ویژن با توجه به خواسته‌های نقشه فنی، دو پایه از PD.۷ را به حالت خروجی و پایه PC.۰ را در حالت ورودی با وضعیت Pullup قرار دهید.



شکل ۷-۳۳ مدار کنترل موتور توسط میکروکنترلر

جدول ۷-۳

برنامه به زبان C برای حالت ۱	برنامه به زبان C برای حالت ۲
<pre> while (۱) { // Place your code here if(PINC.۰==۰){ PORTD.V=۱; } PORTD.V=۰; } </pre>	<pre> while (۱) { // Place your code here if(PINC.۰==۰){ PORTD.V=۱; } else{ PORTD.V=۱; delay_ms(۱۰); PORTD.V=۰; delay_ms(۵); } } } </pre>

- ۴ در حالت اول، در جدول ۷-۳ ستون سمت چپ برنامه به زبان C، در نرم‌افزار کد ویژن طوری نوشته شده است که با فعال کردن کلید موتور روشن شود.
- ۵ تمام مراحل شبیه‌سازی مدار را در نرم‌افزار پروتئوس انجام دهید.
- ۶ برنامه را توسط پروگرامر به میکروکنترلر انتقال دهید.
- ۷ مدار سخت‌افزاری کنترل موتور را روی برد برد ببندید.

اتصال صحیح تغذیه ۵ ولت ثابت به میکروکنترلر و ۱۲ ولت ثابت را به موتور برقرار کنید. مراقب باشید که اتصال کوتاه بین منابع ولتاژ ایجاد نشود. در صورت چنین رخدادی، احتمال آسیب رسیدن به میکروکنترلر وجود دارد.

نکته ایمنی



توجه



روی محور موتور یک پروانه پلاستیکی نصب کنید تا تفاوت سرعت موتور مشخص شود.

۸ چگونگی روشن شدن و تغییرات سرعت موتور را با توجه به فعال کردن کلید فشاری به طور خلاصه توضیح دهید.

.....

.....

.....

- ۹ در حالت دوم، با توجه به جدول ۷-۳ ستون سمت راست، در برنامه تغییرات را به گونه‌ای اعمال کنید، که هنگامی که کلید فشرده نشده سرعت موتور کم و با فشردن کلید سرعت موتور افزایش یابد.
- ۱۰ تمام مراحل شبیه‌سازی مدار را در نرم‌افزار پروتئوس انجام دهید.
- ۱۱ برنامه را توسط پروگرامر به میکروکنترلر انتقال دهید.
- ۱۲ مدار سخت افزاری چشم‌ک‌زن میکروکنترلر را روی بردبرد ببندید.
- ۱۳ کارکرد موتور در دو حالت برنامه‌ریزی میکروکنترلر را باهم مقایسه کنید. تفاوت‌های شروع به کار و سرعت موتور در دو حالت کنترل موتور با میکروکنترلر را به اختصار توضیح دهید.

- ۱۴ نقش دیود D۱ را در مدار کنترل موتور بنویسید.
- ۱۵ برای کنترل دور موتور در این مدار عرض پالس اعمال شده به بیس ترانزیستور تغییر نموده است. (با مشاهده شکل موج بیس ترانزیستور)
- درست ☐ نادرست ☐

- ✓ چگونه می‌توان سرعت موتور را چهار حالتی کرد، به طوری که با هر بار کلید زدن سرعت افزایش پیدا کند.
- ✓ برای تغییر جهت گردش موتور چه مداری پیشنهاد می‌دهید، پیشنهادها را در کارگاه به بحث بگذارید.

بحث کنید



برای مدار شکل ۷-۳۳، دو روش دیگر کنترل دور موتور طراحی کنید و مدار سخت‌افزاری آنها را ببندید.

فعالیت



راه‌اندازی و نمایش اعداد در نمایشگر هفت قطعه‌ای (۷Seg)

هدف: کسب مهارت در برنامه‌نویسی، شبیه‌سازی، پروگرام کردن میکروکنترلر ۸ Atmega و بستن مدار سخت‌افزاری

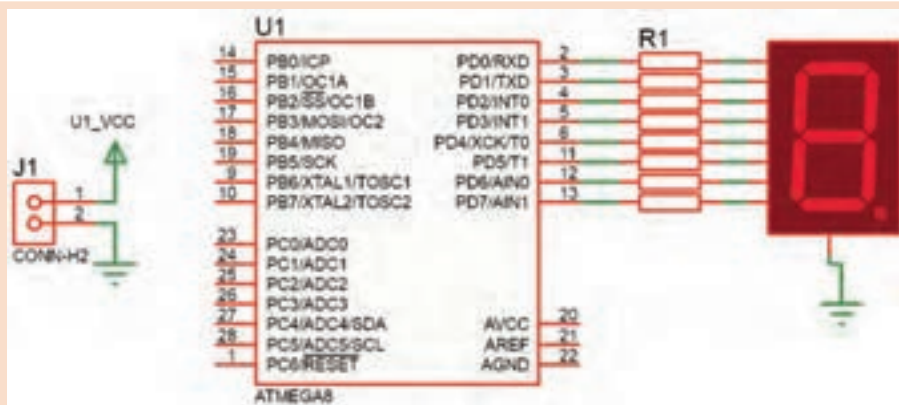
مواد، ابزار و تجهیزات: لوازم التحریر - دستوره‌های زبان C - برگه اطلاعات میکروکنترلر ATmega۸ - دستگاه پروگرامر (Programmer) - رایانه - نرم‌افزارهای مورد نیاز - بردبرد - میکروکنترلر ATmega۸ - میز آزمایشگاهی الکترونیک با تجهیزات استاندارد - مقاومت 220Ω ، $\frac{1}{4}$ وات هشت عدد - سیم‌های بردبرد - هفت قطعه‌ای (۷Seg) کاتد مشترک

مراحل اجرای کار:

- ۱ مدار شکل ۷-۳۴ مدار ساده برای روشن شدن و راه‌اندازی یک نمایشگر هفت قطعه‌ای (۷Seg) سون سگمنت) به کار می‌رود، که می‌تواند با تأخیر ۷۵/۰ ثانیه اعداد ۰ تا ۹ و ممیز را نمایش دهد.
- ✓ در ۷ Seg آند مشترک برای روشن شدن هر LED باید پایه آن را صفر کرد و در سون سگمنت کاتد مشترک برای روشن کردن هر LED باید پایه آن را یک کنیم.

کار عملی ۷





شکل ۷-۳۴ مدار نمایش ارقام با میکرو

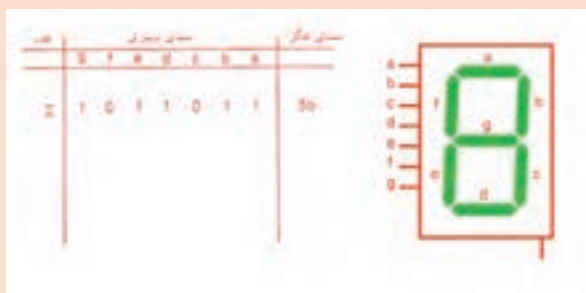
جدول ۷-۴

برنامه به زبان C

```
while (1)
{
// Place your code here
PORTD = 0x3f; //0
delay_ms(750);
PORTD = 0x06; //1
delay_ms(750);
PORTD = 0x5b; //2
delay_ms(750);
PORTD = 0x4f; //3
delay_ms(750);
PORTD = 0x66; //4
delay_ms(750);
PORTD = 0x6d; //5
delay_ms(750);
PORTD = 0x7d; //6
delay_ms(750);
PORTD = 0x07; //7
delay_ms(750);
PORTD = 0x7f; //8
delay_ms(750);
PORTD = 0x6f; //9
delay_ms(750);
PORTD = 0x80; //point
delay_ms(750);
}
}
```

پورتی که ۷Seg به آن متصل می شود را به عنوان خروجی در نظر می گیریم.

برای روشن شدن ۷Seg علاوه بر وارد کردن کدهای باینری می توانید از کدهای هگزا دسیمال نیز استفاده کنید.



شکل ۷-۳۵ جدول کدهای ارقام صفر تا ۹ نمایش در ۷ Seg

برای نمونه در جدول شکل ۷-۳۵ کد باینری و هگزا دسیمال عدد ۲ نوشته شده است. برای سایر ارقام صفر تا ۹ کد باینری و هگزا دسیمال آنها را بنویسید.

۲ مدار شکل ۷-۳۴ را در نرم افزار پروتئوس Proteus ببندید.

۳ ۷Seg برای این پروژه از نوع کاتد مشترک است.

✓ در قسمت Keywords مطابق شکل ۷-۳۶ عبارت ۷seg را جستجو کرده و نوع کاتد مشترک آن را انتخاب کنید.

۲ در نرم افزار کد ویژن با توجه به خواسته های نقشه فنی تمام پایه پورت D را از PD۰ تا PD۷ را در حالت خروجی قرار دهید.

۴ برای حالت اول در جدول ۷-۴ برنامه نشان دادن اعداد ۰ تا ۹ آمده است.

۵ تمام مراحل شبیه سازی مدار را در نرم افزار پروتئوس انجام دهید.

۶ برنامه را توسط پروگرامر به میکرو انتقال دهید.

۷ مدار سخت افزاری نمایشگر ۷Seg با میکروکنترلر را روی برد برد بندید.

۸ چگونگی روشن شدن و سرعت نمایشگر ۷ Seg را مشاهده کنید و به سؤالات زیر پاسخ دهید.
روش نوشتن در پورت D به صورت بیت یا بایت است؟

☞ مفهوم کد x۳f و نوع آن را بنویسید؟ آیا به روش های دیگر نیز می توان نوشت؟

☞ انتهای بعضی خط های برنامه دو خط کج (//) قرار داده شده است، آیا این دستور نمایش اعداد در میکروکنترلر است؟ توضیح دهید.

☞ آیا حروف را در ۷Seg می توان نمایش داد؟ چگونه؟ شرح دهید.

۹ برای حالت های دوم و سوم که خودتان تعریف می کنید، با استفاده از برنامه آماده، مراحل کار را تکرار کنید.

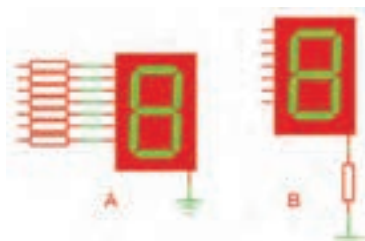


شکل ۷-۳۶ انتخاب ۷ Seg در نرم افزار پروتئوس



با مراجعه به رسانه‌های مختلف، در رابطه با کدهای حروف نمایشگر ۷ Seg تحقیق کنید و نتیجه تحقیق خود را به کارگاه ارائه دهید.

کدام یک از مدارهای شکل ۷-۳۷ را برای استفاده از ۷ Seg پیشنهاد می‌دهید، دلیل خود را ذکر کنید.



شکل ۷-۳۷ انتخاب بایاس ۷Seg

.....

.....

.....

.....

.....

۷-۵ متغیرها در زبان C

یک متغیر محدوده‌ای از فضای حافظه است که با یک نام مشخص می‌شود. یک متغیر بسته به نوع آن می‌تواند حامل یک مقدار عددی باشد. یک متغیر می‌تواند در محاسبات شرکت کند و یا نتیجه محاسبات را در خود حفظ کند. در کل می‌توان گفت که نتایج بخش‌های مختلف یک برنامه، در متغیرها ذخیره می‌شود. در جدول ۷-۵ انواع متغیرها، فضایی که در حافظه اشغال می‌کنند و بازه مقدارپذیری آنها را در کامپایلر کد ویشن مشاهده می‌کنید.

جدول ۷-۵ متغیرها در زبان C

Type	Size (Bits)	Range
bit	1	0 , 1
bool, Bool	8	0 , 1
char	8	-128 to 127
unsigned char	8	0 to 255
signed char	8	-128 to 127
int	16	-32768 to 32767
short int	16	-32768 to 32767
unsigned int	16	0 to 65535
signed int	16	-32768 to 32767
long int	32	-2147483648 to 2147483647
unsigned long int	32	0 to 4294967295
signed long int	32	-2147483648 to 2147483647
float	32	$\pm 1.175e-38$ to $\pm 3.402e38$
double	32	$\pm 1.175e-38$ to $\pm 3.402e38$

■ چگونگی تعریف متغیرها:

متغیرها به صورت زیر تعریف می‌شوند:

؛ مقدار اولیه = نام متغیر نوع متغیر

مثال ۱

Unsigned char A=۱۲;

int a,X, j;

شرح مثال: با توجه به جدول ۵-۷، unsigned char یک متغیر ۸ بیتی است که می تواند بازه ۰ تا ۲۵۵ را داشته باشد.

در خط اول یک متغیر ۸ بیتی بدون علامت با نام A که تنها می تواند مقادیر ۰ تا ۲۵۵ بگیرد، با مقدار اولیه ۱۲ تعریف شده است. در خط دوم نیز ۳ متغیر علامت دار با نام های a و X و z که هر سه مقدار اولیه صفر دارند تعریف شده است که در صورت واگذاری می توانیم مقادیر متفاوتی با توجه به جدول به آن بدهیم.

■ ویژگی های نام متغیر:

- ✓ اولین کاراکتر نام متغیر، نمی تواند عدد باشد.
- ✓ نام متغیر نمی تواند بیشتر از ۳۱ کاراکتر داشته باشد.
- ✓ نام متغیر تنها ترکیبی از حروف a تا z، A تا Z، اعداد و کاراکترها می تواند باشد.

■ انواع متغیرها از نظر محل تعریف در برنامه

- متغیرها از نظر مکانی که در برنامه تعریف می شوند، به دو دسته کلی: متغیرهای عمومی (Global) و متغیرهای محلی (Local) تقسیم می شوند.
- ✓ **متغیرهای عمومی (Global):** متغیرهایی که قبل از تابع main تعریف می شوند را متغیرهای عمومی گویند و در تمام نقاط برنامه می توان به آن دسترسی داشت. تعریف یک متغیر عمومی در ابتدای برنامه، مقدار مشخصی از حافظه را برای همیشه به آن متغیر اختصاص می دهد.
- ✓ **متغیرهای محلی (Local):** متغیرهای محلی در بدنه توابع تعریف می شوند و در بیرون از آن تابع، دسترسی به آن ممکن نیست. متغیرهای محلی تنها، در زمان احتیاج تعریف شده و در حافظه می نشینند و بعد از مدتی از حافظه پاک می شوند.

فیلم مفهوم بیت و بایت و اعداد صحیح و اعداد اعشاری را ببینید.

فیلم



الگوی ارزشیابی نظری (با توجه به جدول ۵-۷ و سایر جداولی که در اختیار دارید به سؤالات پاسخ دهید)

۱ مفهوم متغیر را شرح دهید و کاربرد آن را بنویسید.

۲ مفهوم متغیر بیتی را تعریف کنید. این متغیر با چه کلمه ای معرفی می شود؟

۳ متغیر بایت تا چه عددی ظرفیت دارد و با چه کلمه ای معرفی می شود؟

۴ متغیر int (integer) به چه متغیری گفته می شود؟

۵ تفاوت متغیر بدون علامت و با علامت را بنویسید. کدام یک عدد بزرگ تری را می تواند در خود جای دهد؟

۶ اعداد اعشاری را چگونه معرفی می کنند؟ یک متغیر از نوع اعداد اعشاری، چه مقدار از حجم حافظه را اشغال می کند؟

۷ جدول ۷-۶ را کامل کنید.

جدول ۷-۶

مقدار عددی که باید در متغیری نگهداری شود	متغیر مناسب	متغیر نامناسب	حجم حافظه اشغالی و حداکثر عدد
برای مثال : متغیر بین ۱ تا ۱۵	char unsigned char	int , float , bool	۱ byte, ۰ ... ۲۵۵ ۱ byte , - ۱۲۸ ... ۱۲۷
بین ۱۰ تا ۱۰۵			
برای روشن و خاموش کردن ۱ لامپ			
بین ۶۷- تا ۸۹			
بین ۱۷۸- تا ۱۸۶۷			
بین ۰ تا ۴۰۹۸۷			
بین ۰/۸۹۵ تا ۳/۲۵۴			
برای ایجاد مفهوم شب و روز			

مفهوم گروه متغیرها (آرایه ها یا جدول متغیرها) را بنویسید؟

به نظر شما اگر بخواهیم در کار عملی ۶ یک آرایه برای اعداد ۰ تا ۹ مشخص کنیم، در برنامه C چه تغییری ایجاد می شود، پیشنهاد خود را بنویسید و نتیجه را در کلاس به بحث بگذارید. مشاهده فیلم و مراجعه به کتاب همراه می تواند برای شما کمک کننده باشد.

با مراجعه به رسانه های مختلف و help نرم افزار کدویژن اطلاعاتی در رابطه با سایر متغیرها را استخراج کنید و نتیجه تحقیق خود را ارائه دهید.

فعالیت



تحقیق کنید





مدار شمارنده با (۷ Seg)

هدف: کسب مهارت در برنامه‌نویسی، شبیه‌سازی، پروگرام کردن میکروکنترلر ۸ ATmega و بستن مدار سخت‌افزاری



مواد، ابزار و تجهیزات: لوازم التحریر -

دستورهای زبان C - برگه اطلاعات

میکروکنترلر ۸ ATmega - دستگاه پروگرامر

(Programmer) - رایانه - نرم‌افزارهای مورد

نیاز - بردبرد - میکروکنترلر ۸ ATmega -

میز آزمایشگاهی الکترونیک با تجهیزات

شکل ۷-۳۸ مدار شمارنده یک رقمی

استاندارد - مقاومت 220Ω ، $\frac{1}{4}$ وات هشت عدد - سیم‌های بردبرد - هفت قطعه‌ای (۷ Seg) کاند مشترک

مراحل اجرای کار:

۱ مدار شکل ۷-۳۸ را در نرم‌افزار پروتئوس Proteus رسم کنید.

۲ در نرم‌افزار کد ویژن با توجه به خواسته‌های نقشه فنی، تمام پایه‌ها از PD۰ تا PD۷ را در حالت خروجی قرار دهید و PC۰ را در حالت Pullup و به‌عنوان ورودی در نظر بگیرید.

۳ در نرم‌افزار کدوین شکل ۷-۳۹، ابتدا این دو متغیر را در مکان مشخص شده (مکان متغیرهای عمومی) قرار دهید.

```
#include <mega8.h>
#include <delay.h>

// Declare your global variables here

char NUM[10]={0x3f,0x06,0x5b,0x4f,0x66,0x6d,0x7d,0x07,0x7f,0x6f};
char N;
```

شکل ۷-۳۹ ایجاد آرایه و دادن اطلاعات به آن

جدول ۷-۷

برنامه به زبان C

```
N=0;
while (1)
{
// Place your code here
if(PINC.0==0){
delay_ms(500);
if(PINC.0==0){
N=N+1;
}
if(N>9){
N=0;
}
}
PORTD=NUM[N];
}
```

۴ برنامه‌ای بنویسید که با هر بار فشردن کلید، یک عدد در نمایشگر ۷ Seg اضافه شود و بعد از عدد ۹ به صفر تبدیل شود. در جدول ۷-۷ این برنامه آمده است.

۵ مراحل شبیه‌سازی مدار را در نرم‌افزار پروتئوس انجام دهید.

۶ برنامه را توسط پروگرامر به میکروکنترلر انتقال دهید.

۷ مدار سخت‌افزاری شمارنده با ۷ Seg را بر روی بردبرد ببندید.

۸ مدار را راه‌اندازی کنید. با فشردن کلید شمارش را انجام دهید.

۹ عملکرد مدار شکل ۷-۳۴ را با مدار شکل ۷-۳۸ مقایسه کنید و پرسش‌های زیر را با توجه با جداول مختلفی که در اختیار دارید پاسخ دهید.

مفهوم $N=N+1$ را شرح دهید و این عبارت در برنامه به چه منظور استفاده شده است؟

تفاوت برنامه مدار شکل ۷-۳۸، با برنامه مدار شکل ۷-۳۴ در روش ارسال اطلاعات به پایه‌های ۷Seg را توضیح دهید.

دلیل استفاده این شرط $\{if(N>9)\}$ را بنویسید.

۱۰ به چه دلیلی $delay_ms(500)$ استفاده شده است؟

۱۱ در روش معرفی متغیرها یک متغیر به شکل $NUM[10]=\{ \dots \}$ تعریف شده است. چه نوع متغیری است؟ آیا روش دیگری برای قرار دادن اطلاعات در داخل آن وجود دارد؟ شرح دهید.

۱۲ عدد ۱۰ در متغیر $NUM[10]$ به چه مفهوم است؟

دو روش دیگر برنامه‌نویسی توسط مری برای مدار شمارنده یک رقمی در اختیار شما قرار می‌گیرد. آن را بررسی کنید و مدار سخت‌افزار آن را ببندید.

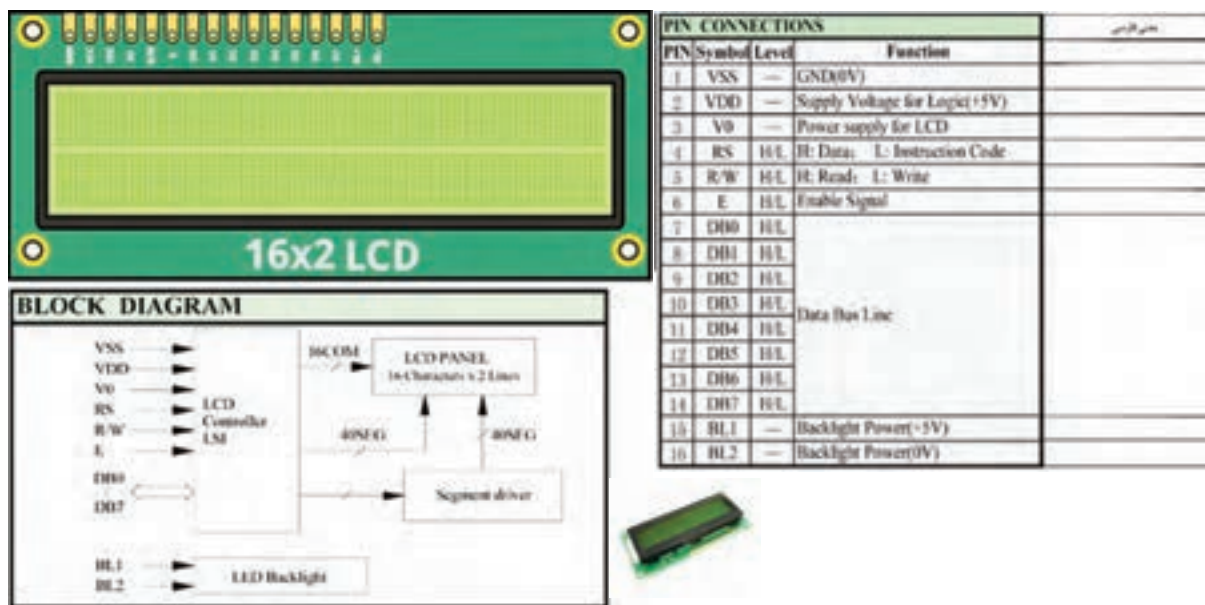
فعالیت



■ صفحه نمایش کاراکتری LCD

صفحه نمایش کاراکتری LCD یکی از پرکاربردترین وسایل خروجی است که به میکروکنترلر وصل می‌شود و مقدار متغیرهای ورودی میکروکنترلر یا پیام‌ها را برای کاربر نمایش می‌دهد. برای این منظور می‌توان کاراکترهای قابل چاپ را روی آن نمایش داد. مشخصه اصلی LCDهای کاراکتری تعداد سطر و ستون آنها است. بلوک دیاگرام، برگه اطلاعات و شکل ظاهری پایه‌های یک LCD 16×2 که دارای ۲ سطر و ۱۶ ستون است را در شکل ۷-۴۰ مشاهده می‌کنید. با این نمایشگر می‌توان در مجموع ۳۲ کاراکتر را نشان داد.

جدول ۷-۸



شکل ۷-۴۰ نمایشگر LCD ۱۶×۲

مدار ساده راه اندازی LCD به صورت نمایش اعداد و متن

هدف: کسب مهارت در برنامه نویسی، شبیه سازی، پروگرام کردن میکروکنترلر ATmega ۸ و بستن مدار سخت افزاری

مواد، ابزار و تجهیزات: رایانه - نرم افزارهای مورد نیاز - دستورهای زبان C - میکروکنترلر ATmega ۸ - دستگاه پروگرامر (Programmer) - سیم های بردبرد - میز آزمایشگاهی الکترونیک با تجهیزات استاندارد - ۱۶×۲ LCD - شارژر موبایل ۵ولتی - مقاومت 560Ω ، $\frac{1}{4}$ وات یک عدد - پتانسیومتر $10 K\Omega$ یک عدد - بردبرد

کار عملی ۹



فیلم آموزشی چگونگی برقراری ارتباط با LCD در نرم افزار کدویژن را ببینید.

فیلم



برای اجرای این کار عملی از برنامه داده شده در جدول ۷-۹ استفاده کنید.

توجه



مراحل اجرای کار

۱ به برگه اطلاعات ۱۶×۲ LCD شکل ۷-۴۰ مراجعه کنید و مفاهیم هریک از کلمات خواسته شده را در جدول ۷-۸ به فارسی ترجمه کنید.

۲ برای آوردن نمایشگر کاراکتری روی میز کار نرم افزار پروتئوس، در قسمت Keywords عبارت lcd را تایپ کنید. سپس LM۰۱۶L که نمایشگر ۱۶×۲ است را مطابق شکل ۷-۴۱ روی میز کار بیاورید.

جدول ۷-۹

برنامه به زبان C

```
while (۱)
{
// Place your code here
lcd_puts("Hello");
delay_ms(۱۰۰۰);
lcd_clear();
lcd_puts("I am a");
delay_ms(۱۰۰۰);
lcd_gotoxy(۰,۱);
lcd_puts("Good Student");
delay_ms(۲۰۰۰);
lcd_clear();
lcd_puts("*** God by ***");
delay_ms(۳۰۰۰);
lcd_clear();
}
}
```

۳ مدار شکل ۷-۴۲ را در نرم افزار پروتئوس رسم کنید.

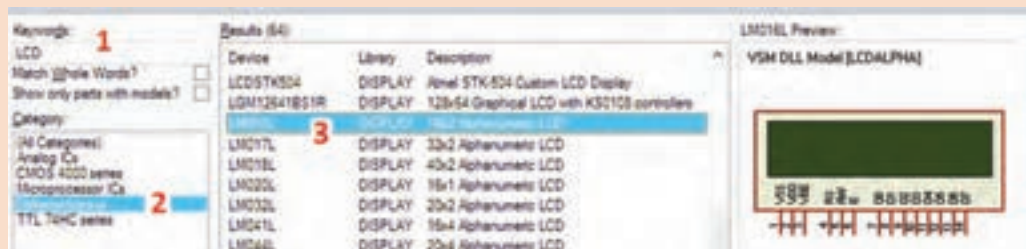
۴ در نرم افزار کد ویژن با توجه به خواسته های نقشه فنی، تمام پایه های مورد نیاز را برای حالت ارتباط با LCD تنظیم کنید، برای این کار بعد از ایجاد پروژه در محیط کدویژن در قسمت ویزارد زبانه Lcd را انتخاب کنید شکل ۷-۴۳.

۵ برنامه داده شده در جدول ۷-۹ را که ابتدا با پیام «Hello» شروع شده و خود را با عنوان یک «دانش آموز خوب» معرفی می کند. در انتها با پیام خداحافظی «Good by» برنامه را تمام می کند، مورد بررسی قرار دهید.

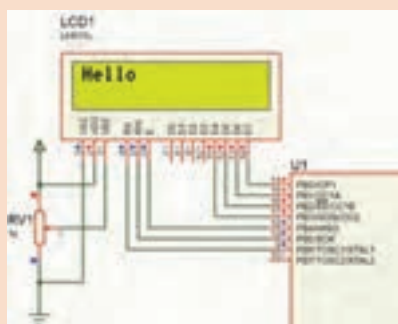
۶ مراحل شبیه سازی مدار را در نرم افزار پروتئوس انجام دهید.

۷ برنامه را توسط پروگرامر به میکروکنترلر انتقال دهید.

۸ مدار سخت افزار نمایشگر LCD را روی بردبرد ببندید.



شکل ۷-۴۱ انتخاب نمایشگر LCD

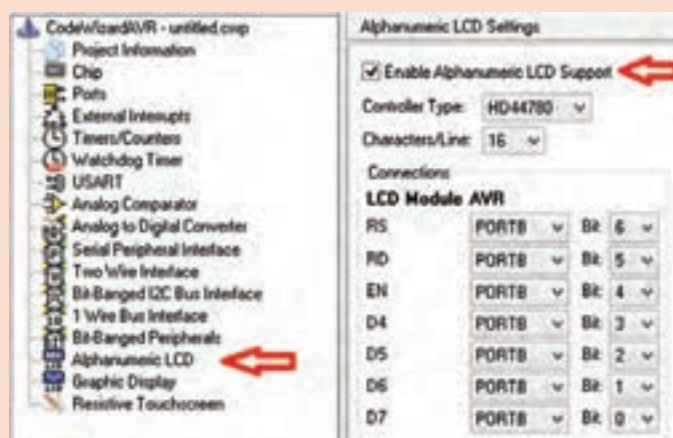
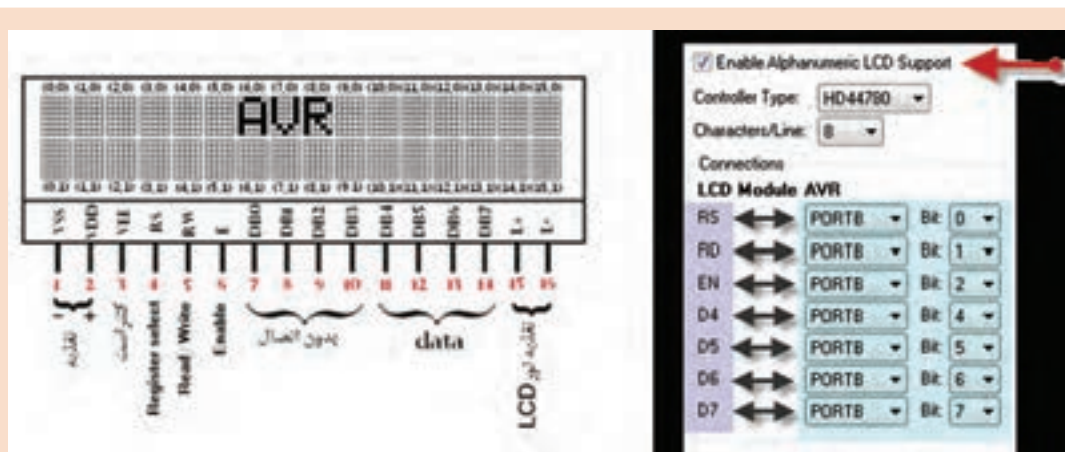


شکل ۷-۴۲ مدار نمایشگر LCD

- هنگام سیم بندی و برقراری اتصال به پایه های LCD ، نهایت دقت را داشته باشید تا نمایشگر LCD آسیب نبیند.
- توجه به شدت نور LCD داشته باشید و با تنظیم پتانسیومتر روشنایی LCD، مقدار نور آن را در حد مناسب قرار دهید شکل ۷-۴۴.

نکته ایمنی





شکل ۷-۴۳ راهنمای تنظیم پایه‌های LCD

۹ مدار را راه‌اندازی کنید. سپس با مراجعه به منابع مختلف به پرسش‌های زیر پاسخ دهید.

۱ دستور lcd_puts چه کاری انجام می‌دهد؟

۲ دستور lcd_clear چه کاربردی دارد؟

۳ دستور lcd_gotoxy چه عملی انجام می‌دهد؟



شکل ۷-۴۴ اتصال صحیح LCD و تنظیم شدت روشنایی

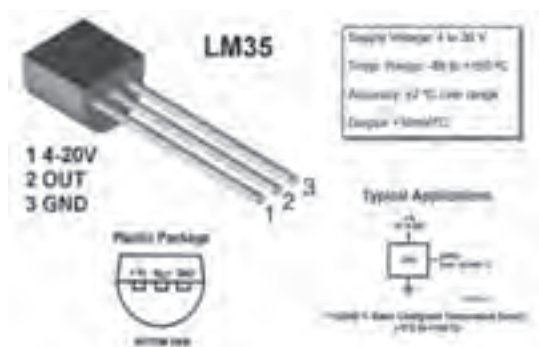


دو برنامه دیگر که برای نمایش متن در نمایشگر LCD است در اختیار شما قرار می‌گیرد، آنها را بررسی کنید و مدار سخت افزار آن را ببندید.

واحد مبدل آنالوگ به دیجیتال A/D

واحد مبدل سیگنال‌های آنالوگ به دیجیتال (ADC = Analog to Digital Converter) مداری الکترونیکی است که سیگنال‌های پیوسته آنالوگ را به داده‌های دیجیتالی تبدیل می‌کند. تمامی کمیت‌های فیزیکی مانند دما، نور، نیرو، فشار، جابه‌جایی و حرکت آنالوگ هستند. برای پردازش توسط میکروکنترلر باید کمیت‌های آنالوگ تبدیل به دیجیتال (۰ و ۱) شوند. عمل تبدیل ولتاژ ورودی آنالوگ به کد دیجیتال متناسب با ولتاژ آنالوگ توسط این واحد انجام می‌شود.

کنترل دمای محیط، کنترل سطح مایع در مخازن (مانند کنترل سطح سوخت خودرو)، کنترل وزن و فشار با میکروکنترلر از متداول‌ترین پروژه‌های کاربردی در صنایع هستند. به همین دلیل مدار کنترل دمای محیط با حسگر حرارتی LM35 در حد گسترده مورد استفاده قرار می‌گیرد.



شکل ۴۵-۳۵ LM حسگر دما

LM35 یک قطعه حس‌کننده حرارت همراه با یک مدار الکترونیکی است که همگی به صورت یک مدار مجتمع ساخته شده‌اند. این مدار به صورت یک قطعه سه پایه شبیه یک ترانزیستور ساخته می‌شود. خروجی این آی سی (IC) به صورت خطی است و به ازای هر درجه سانتی گراد دما ۱۰ میلی ولت ولتاژ در اختیار ما قرار می‌دهد. در شکل ۴۵-۷ برگه اطلاعات LM35 را مشاهده می‌کنید.

مدار ساده، کنترل دما با LM35

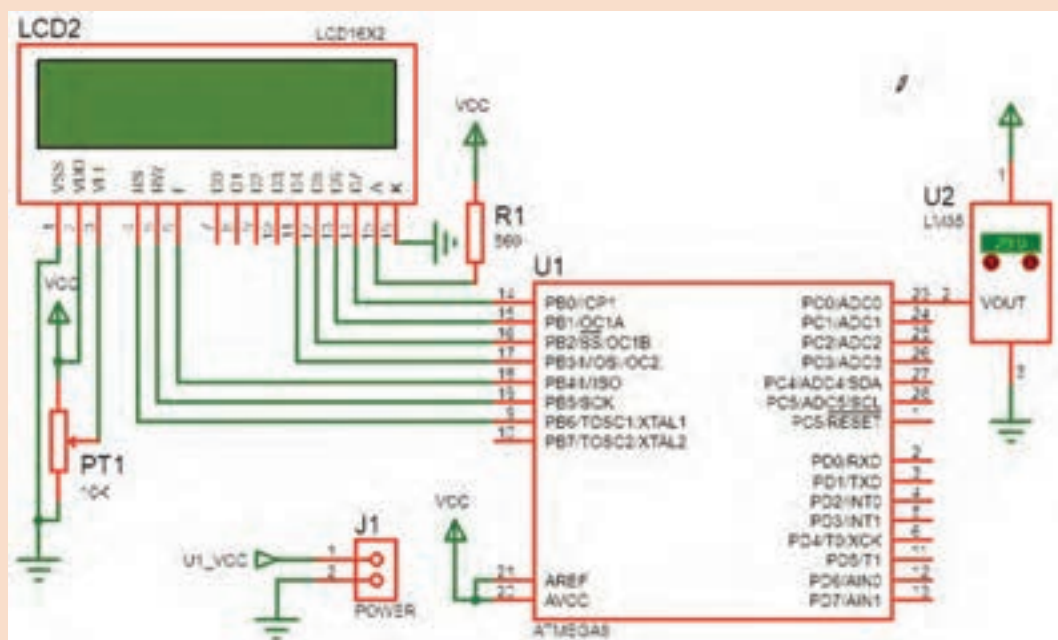
هدف: کسب مهارت در بارگذاری و استفاده از برنامه، شبیه‌سازی، پروگرام کردن میکروکنترلر ۸ ATmega و بستن مدار سخت‌افزاری

مواد، ابزار و تجهیزات: رایانه - نرم‌افزارهای مورد نیاز - دستورهای زبان C - میکروکنترلر ۸ ATmega - دستگاه پروگرامر (Programmer) - سیم‌های بردبرد - میز آزمایشگاهی الکترونیک با تجهیزات استاندارد ۱۶×۲ - LCD - شارژر موبایل ۵ولتی - مقاومت ۵۶۰ Ω، ۱/۴ وات یک عدد - پتانسیومتر ۱۰ KΩ یک عدد - بردبرد - حسگر LM35

مراحل اجرای کار:

- ۱ مدار شکل ۴۶-۷ را در نرم‌افزار پروتئوس Proteus رسم کنید.
- ۲ در نرم‌افزار کد ویژن با توجه به خواسته‌های نقشه فنی تمام پایه‌های مورد نیاز را برای حالت ارتباط با LCD تنظیم کنید.

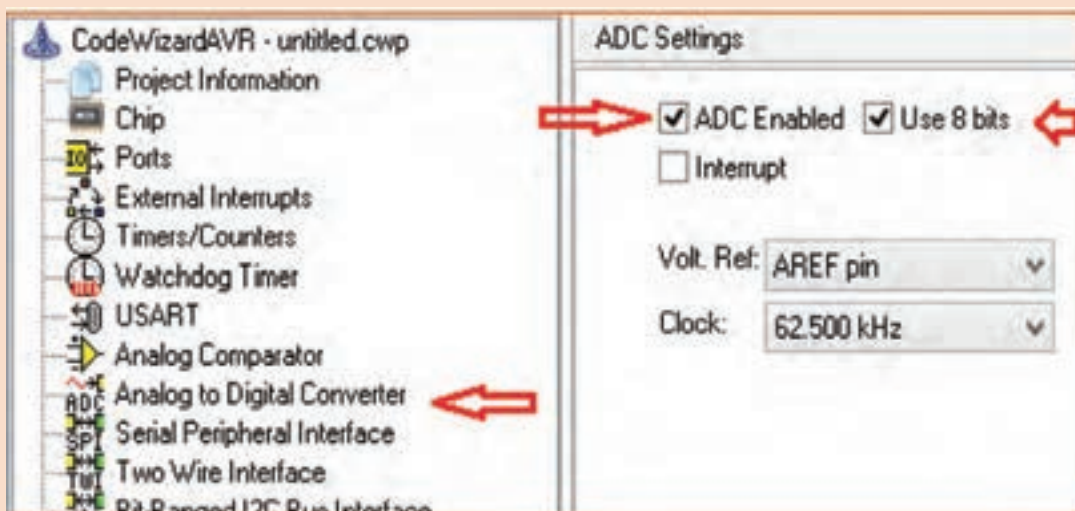




شکل ۷-۴۶ اندازه‌گیری دما و نمایش در LCD

بخش ۳ ADC را در نرم‌افزار کد ویژن فعال کنید، شکل ۷-۴۷.

در کد ویژن در تعریف کتابخانه‌ها، این خط را (`#include <stdio.h>`) اضافه کنید. (این خط برای دستور `sprintf` قرار می‌گیرد)



شکل ۷-۴۷ فعال کردن ADC

جدول ۷-۱۰

برنامه به زبان C

```
while (۱)
{
// Place your code here
A=read_adc(۰);
T=(A*۲۵۵)/۱۳۰;

lcd_clear();
sprintf(PRN,"Tempe= %d",T);
lcd_puts(PRN);
delay_ms(۲۵۰);
}
```

۵ دو متغیر با نام‌های T و A را در بخش تعریف‌های متغیرهای عمومی با عنوان char تعریف کنید. (یک متغیر از نوع آرایه PRN با طول ۱۶ با توجه به طول کاراکتر ۲×۱۶ LCD; char PRN[۱۶];)

دستور `A=read_adc(۰);` از کدام نوع دستورهای استاندارد C، کتابخانه‌ای و تابعی است.

جست و جو کنید



۶ مراحل شبیه‌سازی مدار را در نرم‌افزار پروتئوس انجام دهید.

این برنامه در جدول ۷-۱۰ آمده است.

نکته



۷ با مراجعه به کتاب همراه هنرجو عملکرد مدار را تشریح کنید و به سؤال‌های زیر پاسخ دهید.

۸ دستور `A=read_adc(۰);` چه عملی انجام می‌دهد؟

۹ عملیات ریاضی $T = (A * 255) / 130$ به کار رفته به چه منظور می‌باشد؟

۱۰ دستور `sprintf(PRN, "Temp = %d", T);` چه عملی انجام می‌دهد؟

۱۱ مدار سخت افزاری شکل ۷-۴۶ را با راهنمایی مربی در کارگاه ببینید.



با ترکیب کارهای عملی قبل با این کار عملی حداقل دو مدار کاربردی پیشنهاد کنید.

- Operating Voltages
 - 2.7 - 5.5V (ATmega8L) 12
 - 4.5 - 5.5V (ATmega8)
- Speed Grades
 - 0 - 8 MHz (ATmega8L) 13
 - 0 - 16 MHz (ATmega8)
- Power Consumption at 4 Mhz, 3V, 25°C
 - Active: 3.6 mA 14
 - Idle Mode: 1.0 mA
 - Power-down Mode: 0.5 µA

شکل ۷-۴۸ مقادیر ولتاژ و فرکانس کاری میکرو ۸ ATmega

■ نکات مهم برای استفاده از میکروکنترلر

✓ تغذیه میکرو کنترلرها:

میکروکنترلرها با توجه به نوع قطعات (ترانزیستورهای درون آن) با ولتاژهای متفاوتی کار می‌کنند. ولی معمولاً ولتاژ کار آنها ۵ ولت ثابت DC است ولی مواردی وجود دارد که طبق شکل ۷-۴۸ با ولتاژهای پایین‌تر کار می‌کنند (۱۲ و ۱۳ در شکل ۷-۴۸). همیشه از منبع تغذیه تثبیت شده استفاده کنید. مقدار جریان عبوری در حالت فعال و مصرف کم مشخص شده است. این مقدار مصرف ارتباطی به مقدار جریان مصرفی قطعات جانبی که به میکروکنترلر متصل می‌شوند ندارد.

مثال ۲

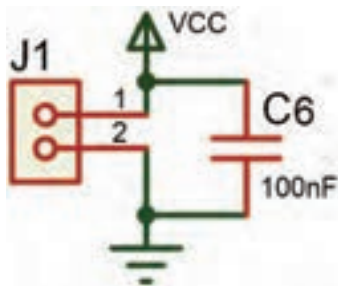
اگر ۱۰ عدد LED که جریان هر کدام ۵ میلی‌آمپر است به میکروکنترلر متصل شوند جمع جریان عبوری از میکروکنترلر و قطعات جانبی آن برابر است با:

$$I_T = (10 \times 5) + 3/6 = 53/6 \text{ mA}$$

میکروکنترلرهای کم مصرف‌تر معمولاً سرعت کمتری دارند. میکروکنترلر ۸ ATmega با ۸L ATmega را در شکل ۷-۴۸ مقایسه کنید.

برای جلوگیری از تولید نویز در قطعات و مدارهای دیجیتال معمولاً همیشه یک خازن ۱۰۰ nF در نزدیکی پایه‌های V_{CC} و GND میکروکنترلر قرار

می‌دهند شکل ۷-۴۹. در نقشه فنی این خازن را در کنار کانکتور تغذیه رسم می‌کنند ولی در ساخت مدار چاپی باید این خازن در کنار پایه‌های تغذیه میکروکنترلر قرار گیرد و به آنها اتصال داده شود.



شکل ۷-۴۹ اتصال خازن ۱۰۰ nF به پایه تغذیه آی‌سی

هم اکنون محصولات این شرکت با پسوند «A» ATmega ۸ A تولید می‌شود که ویژگی‌های هردو محصول قبلی را دارد، یعنی ولتاژ تغذیه ۵/۵ V - ۲/۷ و فرکانس کار تا ۱۶MHz.

نکته



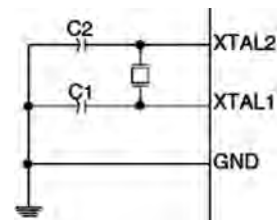
فرکانس کار: برای کم کردن هزینه و سادگی کار با میکروکنترلرهای AVR معمولاً در داخل آن یک اسیلاتور کالبره شده RC قرار می‌دهند. این اسیلاتور با فرکانس‌های (۱، ۲، ۴، ۸) مگاهرتز با دقت بالا کار می‌کند. برای دقت بیشتر می‌توان از اسیلاتور کریستالی نیز استفاده کرد شکل ۷-۵۰. برای این کار نیاز به دو خازن و یک کریستال کوارتز داریم. به این ترتیب در اکثر مدارها نیازی به اسیلاتور کریستالی نیست.

Calibrated Internal RC Oscillator

Table 9. Internal Calibrated RC Oscillator Operating Modes

CKSEL3..0	Nominal Frequency (MHz)
0001 ⁽¹⁾	1.0
0010	2.0
0011	4.0
0100	8.0

CKSEL3=0
CKSEL2=1
CKSEL1=0
CKSEL0=0



خازن و کریستال

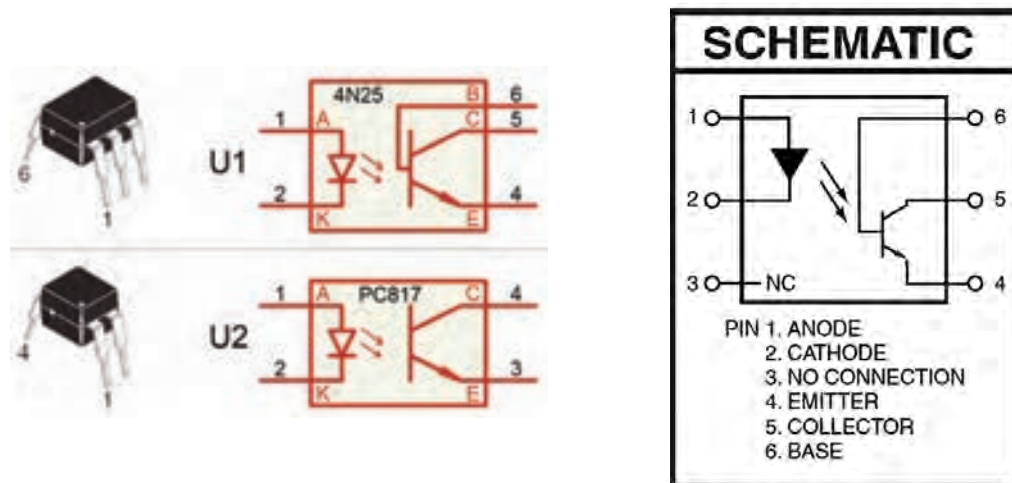
شکل ۷-۵۰ تنظیم فیوز بیت‌ها برای اسبلاتور داخلی RC

✓ **حفاظت از ولتاژ بالا:** هنگامی که از میکروکنترلر برای فرمان دادن به مدارهای ولتاژ بالا استفاده می‌کنید، به حفاظت میکروکنترلر توجه داشته باشید زیرا ولتاژهای بالا می‌توانند به میکروکنترلر آسیب بزنند، یا باعث توقف (Reset) بی‌مورد میکروکنترلر شوند.

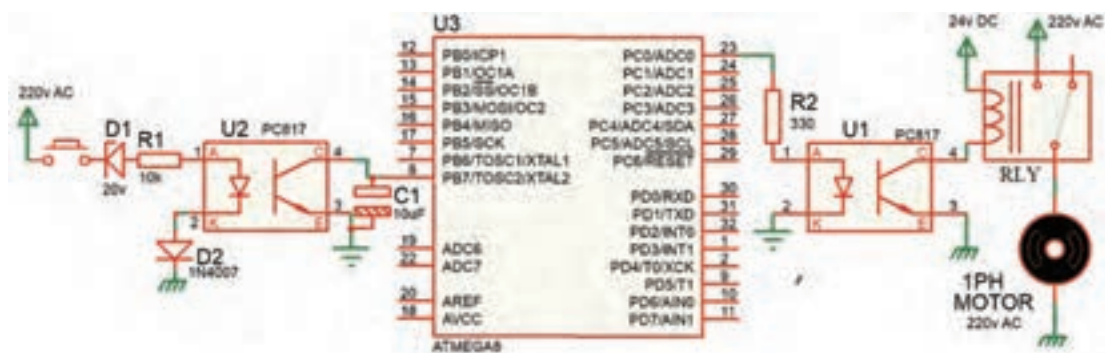
مدارهایی که در آنها سیم پیچ وجود دارد، در زمان القاء به صورت لحظه‌ای ولتاژ بسیار بالایی تولید می‌کنند، در این حالت باید، ارتباط این ولتاژ با ورودی و خروجی میکروکنترلر ایزوله شود. همچنین باید پایه منفی تغذیه از GND ایزوله شود.

برای حفاظت میکروکنترلر روش‌های مختلفی به کار می‌رود. یک روش پرکاربرد و کم هزینه استفاده از قطعه اپتوکوپلر است. نماد فنی و شکل پایه اپتوکوپلر در شکل ۷-۵۱ نشان داده شده است.

در شکل ۷-۵۱ یک LED و یک ترانزیستور بدون هیچ ارتباط الکتریکی و فقط از طریق نور، ترانزیستور را فعال یا غیرفعال می‌کند. در مدل ساده آن که چهار پایه دارد، برای دو وضعیت خاموش و روشن از میکروکنترلر به خروجی یا از ورودی مدار به میکروکنترلر استفاده می‌شود. چگونگی به کارگیری اپتوکوپلر برای حفاظت میکروکنترلر را در شکل ۷-۵۲ مشاهده کنید.

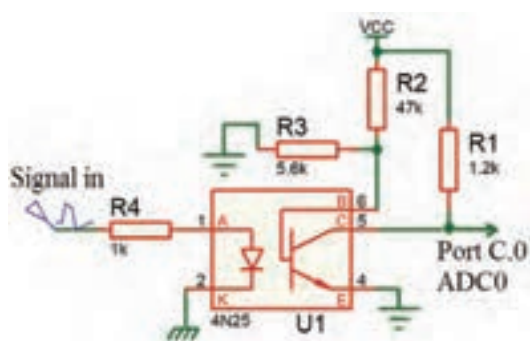


شکل ۷-۵۱ نماد فنی و شکل پایه اپتوکوپلرها



شکل ۷-۵۲ حفاظت میکرو با اپتوکوپلر

در مدار ۷-۵۲ نقش دیود زبر $D1$ کاهش ولتاژ و دیود $D2$ جهت حذف سیکل منفی AC است. مقاومت $R1$ بایاس LED اپتوکوپلر را بر عهده دارد. خازن $C1$ ، خازن صافی است و پالس‌های ایجاد شده در خروجی را به یک ولتاژ DC ساده تبدیل می‌کند. در تنظیم میکروکنترلر، پایه $PB7$ به عنوان کلید ورودی در نظر گرفته شده است. زمانی که ولتاژ $220V$ ولت وجود داشته باشد ولتاژ روی پایه $PB7$ میکروکنترلر کمتر از یک ولت (صفر منطقی) و زمانی که $220V$ AC وجود نداشته باشد، ولتاژ روی پایه $PB7$ میکروکنترلر ۵ ولت (۱ منطقی) می‌شود.



در اپتوکوپلرهای ۶ پایه، پایه بیس B ترانزیستور را می‌توان مطابق شکل ۷-۵۳ بایاس کرد. استفاده از این

اپتوکوپلر دارای این مزیت است که می‌توان حساسیت راه‌اندازی ترانزیستور را تغییر داد و حتی سیگنال‌های کوچک آنالوگ را نیز انتقال داد، مثلاً به آسانی می‌توان ولتاژ خروجی یک حسگر (Sensor) آنالوگ را که ولتاژ خروجی آن در محدوده بسیار کمی قرار دارد به میکروکنترلر انتقال داد.

الگوی پرسش

- ۱ با مراجعه به جداول ۷-۱ تا ۷-۱۰ خطاهای برنامه‌نویسی را در جدول ۷-۱۱ مشخص کنید.
- ۲ در برنامه داده شده در جدول ۷-۱۱ متغیر " NUM[] " اعداد نمایشی برای (۷Seg) آماده شده است. بعد از پایان برنامه در صورتی که (PINC.۰=۰) باشد، عدد نمایش داده شده در (۷Seg) چه خواهد بود؟ با مراجعه به منابع معتبر پاسخ را بیابید.
- ۳ به چه دلیل در برنامه PINC==۰ با فاصله ۱۰۰ میلی ثانیه استفاده شده است؟ با مراجعه به منابع معتبر پاسخ را بیابید.
- ۴ برنامه داده شده برای مدار شکل ۷-۳۴ را بررسی کنید. آیا این برنامه موارد زیر را اجرا می‌کند؟
 - ✓ ابتدا اعداد زوج (۰ تا ۹) را نمایش می‌دهد.
 - ✓ سپس اعداد فرد (۰ تا ۹) را به نمایش در می‌آورد.
 - ✓ در نهایت دوباره از اول اعداد زوج را تکرار می‌کند؟
- ۵ برنامه داده شده برای مدار شکل ۷-۳۴ را بررسی کنید، آیا این برنامه موارد زیر را اجرا می‌کند؟
 - ✓ ابتدا اعداد (۰ تا ۹) را نمایش می‌دهد.
 - ✓ سپس از عدد ۹ به صورت نزولی (۹ تا ۰) به نمایش در می‌آورد.
 - ✓ در ادامه با تأخیر ۵۰۰ میلی ثانیه حالت اول را تکرار می‌کند.
- ۶ در مدار شکل ۷-۵۴ با فرض اینکه مقاومت‌های به کار رفته در مدار ۱k اهمی باشند. مقدار مصرف جریان کل مدار را در بیشترین و کمترین حالت مشخص کنید.
- کمترین = (۷Seg) خاموش، بیشترین = نمایش عدد ۸ + ممیز) از شکل ۷-۵۴ استفاده کنید.
- ۷ در مدار شکل ۷-۵۴ اگر برای افزایش نور، مقاومت‌ها را به ۲۲۰ اهم کاهش دهیم، بیشترین جریان مدار چقدر خواهد شد؟

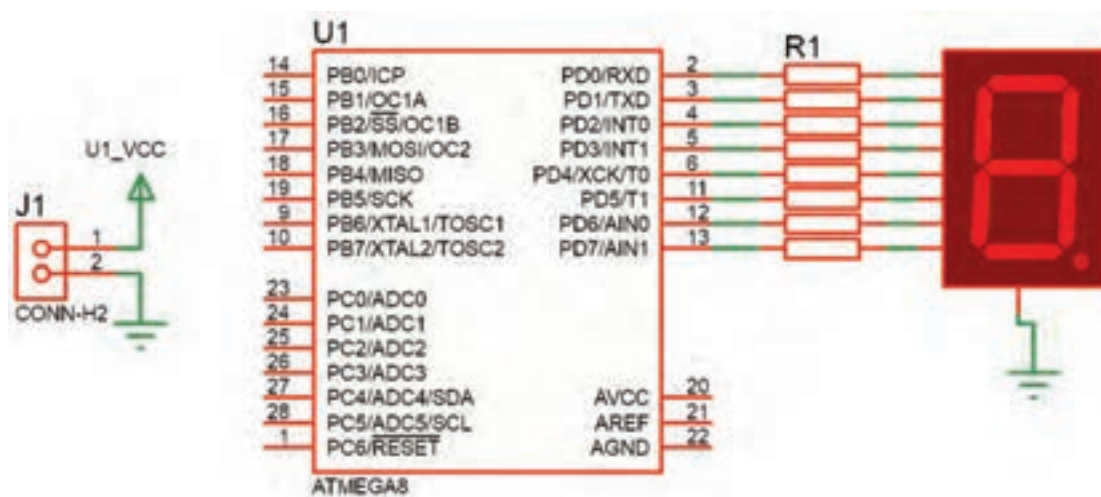
جدول ۷-۱۱

<pre> N=۵; if(PINC.۰==۰){ delay_ms(۱۰۰) if(PINC.۰==۰){ N=N+۱; } if(N>۹){ N=۰; } { PORTD=NUM[N]; } } </pre>	<pre> PORTB.۰=۱; PORTB=۰; Delay_ms(۱۰۰۰); PORTB.۰=۰; PORTB.۱=۱; delay-ms (۱۰۰۰); </pre>
---	--

- ۸ اگر بخواهیم سه عدد هفت قطعه‌ای (۷Seg) به پورت‌های PC، PB و PD متصل کنیم و مقاومت‌ها را ۳۳۰ اهم قرار دهیم حداکثر جریان مصرفی مدار چه مقدار است؟
- ۹ در صورتی که ۷Seg را بخواهیم با مدل آند مشترک عوض کنیم چه تغییری در سخت‌افزار و چه تغییری در نرم‌افزار باید انجام شود؟ توضیح دهید.

برای نمایش حروف J , C , L , U , E و H در شکل ۷-۵۴، چه کدهایی در مبنای هگزادسیمال باید به پورت D ارسال شود؟

تحقیق کنید



شکل ۷-۵۴

نکته



برنامه‌های مربوط به این پروژه‌ها توسط مربی در اختیار هنرجو قرار می‌گیرد.

آزمون عملی پایان واحد یادگیری:

آزمون عملی ۱

اجرای پروژه چشمک‌زن ۸ کانالی (LED ۸) با دو نوع حرکت و چرخش

توجه



این آزمون با برنامه آماده که در اختیار هنرجو قرار دارد اجرا می‌شود.

- ۱ رسم نقشه مدار در نرم‌افزار پروتئوس
- ۲ بارگذاری و اجرای شبیه‌سازی برنامه برای حرکت اول
- ۳ بارگذاری و اجرای شبیه‌سازی برنامه برای حرکت دوم
- ۴ اضافه کردن یک کلید جهت انتخاب حالت ۱ یا ۲ در مدار
- ۵ ترکیب دو برنامه و کلید و اجرا در شبیه‌سازی
- ۶ پروگرام کردن میکروکنترلر و اجرا روی بردبرد

آزمون عملی ۲

اجرای پروژه نمایش اعداد در دو عدد ۷Seg با شمارش از ۱ تا ۹۹

توجه



این آزمون با برنامه آماده که در اختیار هنرجو قرار دارد اجرا می‌شود.

- ۱ رسم نقشه مدار در نرم‌افزار پروتئوس
- ۲ بارگذاری برنامه و اجرای شبیه‌سازی برنامه برای نمایش ۱ تا ۹۹
- ۳ اضافه کردن دو کلید به نقشه و برنامه جهت ایجاد یک شمارنده صعودی نزولی
- ۴ نوشتن برنامه همراه دو کلید به گونه‌ای که با زدن یک کلید یک رقم به نمایش اضافه شود و با زدن کلید دیگر یک رقم کم شود و اجرا شبیه‌سازی در نرم‌افزار پروتئوس
- ۵ پروگرام کردن میکروکنترلر و پیاده‌سازی مدار روی بردبرد

آزمون عملی ۳

اجرای و ساخت یک ولت‌متر و نمایش در LCD

- ۱ رسم نقشه مدار در پروتئوس با مقادیر مناسب قطعات
- ۲ اجرای یک ولت‌متر برای محدوده ۰ تا ۵ ولت و نمایش در LCD
- ۳ اضافه کردن یک کلید جهت انتخاب دو حالت (۰ تا ۵) یا (۰ تا ۲۰) ولت و قطعات مورد نیاز به تشخیص خود هنرجو
- ۴ پروگرام کردن میکروکنترلر و پیاده‌سازی مدار روی بردبرد

آزمون عملی ۴

اجرا و ساخت یک کنترل دور موتور DC توسط سه کلید

- ۱ رسم نقشه مدار در پروتئوس با مقادیر و قطعات مناسب
- ۲ اجرای پروژه چپ‌گرد و راست‌گرد توسط دو کلید (خاموش - چپ‌گرد - راست‌گرد)
- ۳ اجرای کنترل دور موتور به این صورت که (کلید ۱ برای خاموش و روشن شدن - کلید ۲ برای چپ‌گرد و راست‌گرد - کلید ۳ برای حرکت با دو نوع سرعت ۱/۲ قدرت - با قدرت کامل)
- ۴ پروگرام کردن میکروکنترلر و پیاده‌سازی مدار روی بردبرد

ارزشیابی شایستگی

شرح کار:

- ۱ انتخاب میکروکنترلر مناسب (AVR)
- ۲ برنامه‌نویسی میکروکنترلر (دو نمونه برنامه ساده)
- ۳ استفاده از پروگرامر برای برنامه‌ریزی میکروکنترلر
- ۴ استفاده از میکروکنترلر برنامه‌ریزی شده در مدار

استاندارد عملکرد:

برنامه‌ریزی میکروکنترلر و استفاده از آن برای بستن مدار ساده و راه‌اندازی مدار با رعایت قوانین و استاندارد حاکم شاخص‌ها:

- ۱ انتخاب مناسب میکروکنترلر AVR و استخراج اطلاعات مهم از برگه‌های اطلاعاتی (۲۰ دقیقه)
- ۲ شرح چگونگی برنامه‌نویسی ساده (در حد چراغ چشمک زن) میکروکنترلر و نحوه پروگرام کردن برنامه نوشته شده در میکروکنترلر (۳۰ دقیقه)
- ۳ بستن مدار ساده میکروکنترلر برنامه‌ریزی شده (۲۰ دقیقه)

شرایط انجام کار و ابزار و تجهیزات:

شرایط: مکان انجام کار با کف عایق یا آنتی استاتیک - نور مناسب برای کارهای مطالعه و کارهای آزمایشگاهی ساده - دمای طبیعی (۱۸ تا ۲۷ درجه سانتیگراد) و مجهز به وسایل اطفاء حریق - میز کار استاندارد برای رایانه و میز آزمایشگاه الکترونیک با **ابزار و تجهیزات:** رایانه متصل به شبکه اینترنت - فرد با لباس آزمایشگاهی - انجام کار در حال نشسته .
نرم افزارهای مورد نیاز: بریدر - میکروکنترلر ۸ Atmega - دستگاه پروگرامر (Programmer) - دیود LED قرمز و سبز - مقاومت‌های $560\ \Omega$ - $470\ \Omega$ - $220\ \Omega$ - $3/3\ K\Omega$ ، $1/4$ وات ترانزیستور BC140 - سیم‌های بریدر 16×2 - LCD - کاتد مشترک - موتور DC کوچک - کلید شستی کوچک (SPST Push Button) - شارژر موبایل ۵ ولتی

معیار شایستگی:

ردیف	مراحل کار	حداقل نمره قبولی از ۳	نمره هنجار
۱	انتخاب میکروکنترلر مناسب (AVR)	۱	
۲	برنامه‌نویسی میکروکنترلر (دو نمونه برنامه ساده در حد چراغ چشمک زن)	۲	
۳	نحوه پروگرام کردن برنامه نوشته شده در میکروکنترلر	۲	
۴	استفاده از میکروکنترلر برنامه‌ریزی شده در مدار	۲	
	شایستگی‌های غیرفنی، ایمنی، بهداشت، توجهات زیست محیطی و نگرش: ۱- رعایت نکات ایمنی دستگاه‌ها ۲- دقت و تمرکز در اجرای کار ۳- شایستگی تفکر و یادگیری مادام‌العمر ۴- اخلاق حرفه‌ای	۲	
میانگین نمرات			*

* حداقل میانگین نمرات هنجار برای قبولی و کسب شایستگی، ۲ می‌باشد.