

تدریس پودمان دوم برنامه نویسی میکروکنترلر (واحد یادگیری او ۲)

تدریس پودمان دوم: برنامه نویسی به زبان C

واحد یادگیری ۳: طراحی الگوریتم (فلوچارت) مدار پروژه ساده الکترونیکی
واحد یادگیری ۴: برنامه نویسی به زبان C و تحلیل برنامه های آماده
پروژه های الکترونیکی

استاندارد تربیت و یادگیری شایستگی
دانش فنی تخصصی - پایه ۱۲ - شاخه فنی و حرفه ای

وزارت آموزش و پرورش
سازمان پژوهش و برنامه ریزی آموزشی
دفتر تألیف کتاب های درسی فنی و حرفه ای و کار دانش



تجهیزات آموزشی: وایت برد - رایانه - برنامه های سامانه های کنترلی	تعداد جلسه: ۷ جلسه	هدف کلی درس: ایجاد چارچوبی مفهومی، ادراکی و عملکردی در ساختار فکری هنجاریان به منظور سازگاری فناورانه و تعمیم دانش، مهارت و شایستگی های حرفه ای در موقعیت های جدید کاری و حرفه ای در دوران زندگی
ساعت آموزش پودمان: ۳۰ ساعت	مکان اجرای آموزش: کلاس درسی مجهز به رایانه و ویدئو پروژکتور	پودمان دوم: برنامه نویسی میکروکنترلر
ارزشیابی: سنجش مبتنی بر شایستگی	زمان پیشنهادی آموزش: (غیردرسی) ۲+۲۸	وسعت محتوا: به کارگیری دستورهای برنامه نویسی زبان C (یا هر زبان به روز دیگر) و نوشتن چند برنامه برای مدارهای الکترونیکی ساده و تحلیل برنامه های آماده در سامانه های کنترلی

پیامدهای یادگیری:

اهداف توانمندساز	جلسه آموزشی	شایستگی‌ها
طراحی الگوریتم برنامه (فلوچارت) و علائم استاندارد طراحی	جلسه اول	طراحی الگوریتم (فلوچارت) مدار پروژه ساده الکترونیکی
بررسی خصوصیات سخت‌افزار و نرم‌افزار پروژه - مفاهیم اولیه زبان C (یا هر زبان به‌روز دیگر)		
انواع داده و متغیر در زبان C (یا هر زبان به‌روز دیگر)		
عملگرها در زبان C (یا هر زبان به‌روز دیگر)		
ساختار کلی برنامه‌نویسی به زبان C (یا هر زبان به‌روز دیگر) - تبدیل مبنا و علائم آن در زبان C (یا هر زبان به‌روز دیگر)	جلسه دوم	
نوشتن یک برنامه ساده به زبان C (یا هر زبان به‌روز دیگر)		
نوشتن دستورهای شرطی، حلقه و حلقه شرطی		
نوشتن چند برنامه ساده با دستورهای حلقه، شرطی و حلقه شرطی در زبان C (یا هر زبان به‌روز دیگر)	جلسه سوم	
تشریح توابع در زبان C (یا هر زبان به‌روز دیگر)		
تشریح توابع کتابخانه‌ای استاندارد		
نوشتن چند برنامه با مثال ساده با زبان C (یا هر زبان به‌روز دیگر)		
حل چند نمونه تمرین و رفع اشکال و ارزشیابی		

تشریح برنامه به زبان C (یا هر زبان به روز دیگر) برای مدار الکترونیکی با چند LED	جلسه چهارم	برنامه نویسی به زبان C (یا هر زبان به روز دیگر) و تحلیل برنامه های آماده پروژه های الکترونیکی
تحلیل برنامه نوشته شده برای شمارنده حلقوی		
تحلیل برنامه نوشته شده برای شمارنده صعودی و نزولی با کلید		
تحلیل برنامه نوشته شده با LCD		
تحلیل برنامه نوشته شده برای یک ساعت دیجیتال همراه با کلید تنظیم ساعت و دقیقه	جلسه پنجم	
تشریح برنامه به زبان C (یا هر زبان به روز دیگر) برای مدار با صفحه کلید و LCD		
تحلیل برنامه نوشته شده برای قفل رمز		
رفع خطا در یک برنامه نوشته شده و ارزشیابی		
تشریح برنامه به زبان C (یا هر زبان به روز دیگر) برای سامانه های کنترلی نور	جلسه ششم	
تشریح برنامه نوشته شده برای سامانه کنترلی دما همراه با کنترل یک فن (Fan)		
جمع بندی و اشاره به سامانه های دیگر		
ارزشیابی و رفع اشکال		
ارزشیابی پودمان دوم		جلسه هفتم

توصیه‌های کاربردی در ارتباط با تدریس پودمان دوم

هدف از تدریس این پودمان آشنایی بیشتر هنرجویان با الگوریتم و برنامه‌نویسی میکروکنترلر و کاربرد آن است. در پایه یازدهم تاحدودی با میکروکنترلر کار کرده‌اند و با آن آشنایی دارند. در این پودمان بر رفتار عملگرها، انواع داده‌ها و متغیرها در زبان C، ساختار کلی برنامه C و نوشتن چند نمونه برنامه ساده خواهید داشت. در واحد یادگیری دوم، به تشریح و تحلیل چند برنامه کاربردی مانند قفل رمز بپردازید. بدیهی است با توجه به بازه گسترده زبان‌های برنامه‌نویسی، نمی‌توان آموزش کامل را از طریق یک پودمان به اجرا در آورد، ولی مقدمه‌ای برای آموزش‌های بعدی هنرجویان در مقاطع بالاتر خواهد بود با توجه به کسب شایستگی‌های هدف‌گذاری شده در برنامه درسی، توصیه می‌شود هنرآموزان برای تدریس هرچه بهتر این پودمان موارد پیشنهادی زیر را در نظر داشته باشند:

■ قبل از شروع به تدریس، تسلط کامل در زمینه محتواهای ارائه شده داشته باشید، تا بتوانند اطلاعات مورد نیاز را به هنرجویان انتقال داده و سؤالات احتمالی آنان را به درستی پاسخ دهید.

■ انتخاب و توصیه برنامه‌های مشابه با توجه به سایر کتاب‌های درسی و آموخته‌های سال قبل و پیشنهاد آن توسط هنرجویان و تأیید آن توسط هنرآموز می‌تواند در فرایند کیفی آموزش مؤثر باشد.

■ برنامه‌های انتخابی باید کاملاً کاربردی، تخصصی، مرتبط با محتوای کتاب‌های درسی پایه‌های دهم، یازدهم و دوازدهم متناسب با سطح علمی هنرجویان رشته الکترونیک باشد.

■ با مراجعه به سایت‌های مختلف داخلی و خارجی می‌توانید محتواهای آموزشی مورد نیاز خود را بیابید و در اختیار هنرجویان قرار دهید و آنها را با سایر هنرآموزان به اشتراک بگذارید.

■ ضرورت دارد که به هنرجویان تذکر داده شود که قبل از شروع به یادگیری محتوا، مروری بر بخش میکروکنترلر در پایه یازدهم داشته باشند.

■ استفاده از برنامه‌های سنگین، مبهم و پیچیده مجاز نبوده و توصیه نمی‌شود.

■ در ابتدای کار ممکن است کار برای هنرجویان کمی سخت باشد، ولی پس از تمرین‌های کافی مشکل حل خواهد شد.

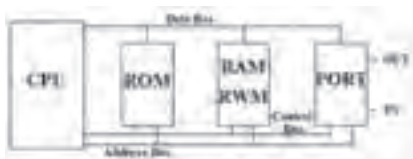
■ توصیه اکید می‌شود که اجرای فعالیت‌های مربوط به این پودمان به صورت گروهی به اجرا در آید و برخی از قسمت‌ها را به صورت کنفرانس ارائه دهند.

■ می‌توانید بخش‌های مختلف متون درسی را در جلسات کلاسی بین گروه‌ها تقسیم کنید و از هر گروه، بخواهید آنها را ارائه دهند، در خلال آموزش اشکال‌های آنان را بر طرف کنید. به این ترتیب یک کلاس تعاملی فعال خواهید داشت.

■ پس از اتمام این پودمان، هنرجو باید موارد صفحه بعد را فراگرفته باشد:

- ✓ الگوریتم برنامه (فلوچارت) را با استفاده از علائم استاندارد، طراحی کنند.
- ✓ انواع داده، متغیر و عملگرها را به کار ببرند.
- ✓ ساختار کلی برنامه‌نویسی به زبان C و اجزای آنها را تشخیص دهند.
- ✓ چند برنامه ساده با دستورهای حلقه، شرطی و حلقه شرطی را در زبان C بنویسند.
- ✓ چند برنامه به زبان C برای مدار الکترونیکی با چند LED را تشریح کنند.
- ✓ چند برنامه نوشته شده برای شمارنده صعودی و نزولی را با کلید و قفل رمز تحلیل کنند.
- ✓ چند برنامه به زبان C برای سامانه‌های کنترلی نور، دما و سامانه‌های دیگر را شرح دهند.

■ اجزای یک سیستم میکروپروسسوری: معمولاً یکی از سؤالاتی که ذهن هنرجویان را درگیر می‌کند، تفاوت میکروپروسسور و میکروکنترلر است. در شکل ۱ بلوک دیاگرام یک سیستم میکروپروسسوری نشان داده شده است.



شکل ۱

✓ CPU (Central Processing Unit): در یک سیستم میکروپروسسوری، CPU کار کنترل سیستم و پردازش داده‌ها را برعهده دارد. برای نمونه می‌توان از پردازنده‌های ۸۰۸۰، ۸۰۸۵، ۸۰۸۶، Z۸۰ و ۶۸۰۰ نام برد. توجه داشته باشید که CPU هر چقدر هم که قوی باشد به تنهایی قادر به انجام کاری نیست و باید همراه با RAM، ROM و PORT در یک سیستم قرار گیرد تا بتواند برنامه‌های نوشته شده را اجرا نماید.

از هنرجویان بخواهید درباره سرعت و تعداد بیت آدرس و دیتای پردازنده‌های نام برده شده تحقیق کنند.

تحقیق



✓ ROM (Read Only Memory): حافظه‌ای است فقط خواندنی، که اطلاعات راه‌اندازی سیستم که به آن سیستم عامل (Operating System - OS) نیز گفته می‌شود، در آن قرار می‌گیرد.

سؤال: چرا لازم است که سیستم عامل در حافظه فقط خواندنی باشد؟

✓ RWM (Read Write Memory): حافظه‌ای با قابلیت خواندن و نوشتن که

اطلاعات میانی سیستم درون آن قرار می‌گیرد و به آن حافظه RAM نیز می‌گویند. توجه داشته باشید که RAM مخفف Random Access Memory است و بیانگر نوع دسترسی به حافظه است. نام صحیح این حافظه، RWM می‌باشد و RAM یک اشتباه مصطلح است.

✓ **PORT**: در هر سیستم تعدادی پورت (PORT) وجود دارد که وظیفه آنها ورود و خروج اطلاعات است.

✓ **BUS**: در هر سیستم سه نوع BUS وجود دارد که عبارت‌اند از: Control, Data و Address

□ **Address** مشخص می‌کند با کدام حافظه یا پورت کار داریم.

□ **Control** مشخص می‌کند چه کاری داریم.

□ **Data** توسط این خطوط اطلاعات بین CPU و بقیه مدار منتقل می‌شود.

✓ به عنوان نمونه می‌توانید برد آزمایشگاهی MPF۱ یا تصویر آن را نمایش دهید.

■ میکروکنترلر (Microcontroller):

اگر یک سیستم میکروپروسسوری را درون یک چیپ قرار داده و فقط پورت‌ها، در اختیار کاربر قرار گیرد به آن میکروکنترلر گفته می‌شود. سیستم‌های میکروپروسسوری دارای حجم زیاد، قیمت زیاد و از همه مهم‌تر مصرف بالا می‌باشند که مانع استفاده از آنها در اکثر کاربردهای کنترلی می‌شود. این محدودیت‌ها در میکروکنترلرها کمتر شده و کاربری آنها در اکثر مدارهای کنترلی را افزایش داده است. یکی از اولین میکروکنترلرهایی که به‌صورت تجاری وارد بازار شد، میکروکنترلر ۸۰۵۱ ساخت شرکت اینتل بوده است.

تحقیق



از هنرجویان بخواهید در مورد بلوک دیاگرام داخلی، ظرفیت حافظه، تعداد پورت‌ها و سرعت میکروکنترلر ۸۰۵۱ تحقیق نمایند.

✓ **ساختار RISC و CISC**: میکروکنترلرها با دو ساختار RISC و CISC ساخته می‌شوند.

□ **RISC**: Reduced Instruction Set Computer

□ **CISC**: Complex Instruction Set Computer

ساختار RISC دارای دستورهای ساده‌تر، برنامه‌نویسی مشکل و سرعت بالا است، (مانند میکروکنترلرهای AVR).

ساختار CISC دارای دستورهای پیچیده، برنامه‌نویسی ساده‌تر و سرعت پایین است، (مانند میکروکنترلرهای ۸۰۵۱).

✓ **رجیسترها**: در هر میکروکنترلر تعدادی رجیستر همه منظوره General Purpose Register وجود دارد، (R_0 تا R_{31}).

همچنین تعدادی رجیستر تک منظوره IO (Special Purpose Register) وجود دارد. وظیفه رجیسترهای IO راه‌اندازی، تنظیم و پیکربندی قسمت‌های مختلف میکروکنترلر است.

تعدادی از رجیسترهای مهم عبارت‌اند از:

DDRA (Data Direction Register - DDR): رجیستر تنظیم پورت A.

PORTA: رجیستر خروجی پورت A.

PINA: رجیستر ورودی پورت A.

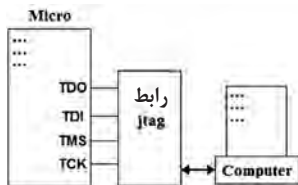
TCCR (Timer Counter Control Register - TCCR): تنظیم‌کننده تایمر کانتر صفر.

✓ **سرعت:** این IC می‌تواند تحت فرکانس ۱۶MHZ با سرعت ۱۶MIPS عملیات را انجام دهد، (MIPS - Million Instruction Per Second)، یعنی در هر ثانیه ۱۶ میلیون دستور را اجرا نماید.



شکل ۲

✓ **حافظه (Memory):** در این میکروکنترلر سه نوع حافظه وجود دارد. در شکل ۲ دیتای ذخیره شده در هر حافظه مشخص شده است.



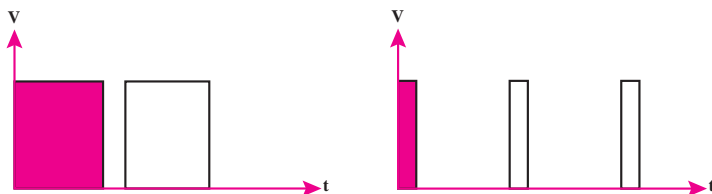
شکل ۳

✓ **(Joint Test Action Group) JTAG:**

یک واسطه بین میکروکنترلر و کامپیوتر است که می‌توان از طریق آن علاوه بر برنامه‌ریزی میکروکنترلر، برنامه داخل میکروکنترلر را خط به خط اجرا و در صورت نیاز آن را تصحیح کرد.

✓ **RTC (Real Time Counter):** اگر بخواهیم یک زمان‌سنج واقعی داشته باشیم، لازم است که بتوانیم پالس یک ثانیه را تولید کنیم. در این میکروکنترلر واحدی به نام RTC وجود دارد که می‌توان توسط آن پالس یک ثانیه و به دنبال آن دقیقه و ساعت را تولید کرد.

✓ **(Pulse Width Modulator) PWM:** یکی از راه‌های کنترل بعضی از دستگاه‌ها، کنترل توانی است که به آنها ارسال می‌شود. توسط تغییر در عرض پالس، می‌توانیم توان ارسالی به دستگاه را کنترل کنیم. برای مثال اگر دو موج هم‌دامنه و هم‌فرکانس مانند شکل ۴ را به دو موتور مشابه بدهیم سرعت آنها با هم متفاوت خواهد بود.



شکل ۵

شکل ۴

در ATmega8 سه کانال PWM پیش‌بینی شده است، (پایه‌های OC2, OC1A, OC1B).

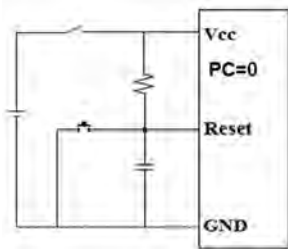
☑ ارتباط: یکی از مسائلی که در کار با میکروکنترلرها با آن مواجه خواهید شد، ارتباط دو یا چند دستگاه است تا بتوان اطلاعات را از یک نقطه به نقطه دیگر منتقل کرد. در این میکروکنترلر سه نوع ارتباط پیش‌بینی شده است.

جدول ۲۲- انواع ارتباط و پایه‌های مربوطه

نوع ارتباط	پایه‌ها	عملکرد پایه
USART	TXD	ارسال دیتا
	RXD	دریافت دیتا
	XCK	پالس همزمانی
I2C (TWI)	SDA	ارسال و دریافت دیتا
	SCL	پالس همزمانی
SPI	MOSI	Master Out Slave In
	MISO	Master In Slave Out
	SCK	Slave Clock
	SS	Slave Select

✓ **WD (Watchdog Timer):** هرگاه سیستمی هنگ کند، برای اینکه دوباره به کار بیفتد لازم است که آن را ریست کنیم. WD وظیفه دارد در صورتی که CPU به هر علتی هنگ کرد آن را ریست نماید. این بخش یک تایمر است که حداکثر تا ۲ ثانیه تنظیم می‌شود. اگر به هر علتی میکروکنترلر هنگ کند و نتواند WD را ریست کند بعد از زمان تنظیم شده WD میکرو را ریست می‌کند.

✓ **Reset:** تمام میکروکنترلرها باید پس از روشن شدن ریست شوند تا کار محول شده را از خط اول برنامه شروع کنند. چگونگی این کار در شکل ۶ نشان داده شده است. در



شکل ۶

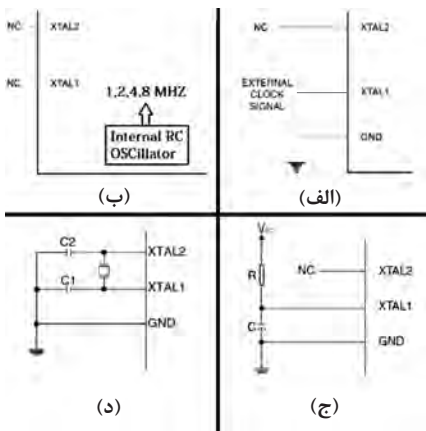
این مدار، به محض اتصال ولتاژ تغذیه، میکروکنترلر ریست می‌شود. به این ویژگی Power On Reset گفته می‌شود. هر وقت میکروکنترلر ریست شود، شمارنده برنامه (Program Counter - PC) برابر صفر شده و در نتیجه برنامه حتماً از خط اول شروع خواهد شد.

✓ **Brown-Out Detection:** سازوکاری است که ولتاژ تغذیه میکروکنترلر را پایش می‌کند تا در صورتی که از مقدار مشخصی کمتر شد، آن را ریست کند.

نکته: در صورت داشتن وقت کافی می‌توانید کاربرد RTC، PWM را با دو مثال به صورت بلوک دیاگرامی، نحوه تولید ساعت و کنترل موتور به روش PWM توضیح دهید. همچنین نقشه مداری را بدهید و از آنان بخواهید که برنامه‌های آن را از سایت‌های مرتبط بارگیری کنند و یک نمونه را با نرم افزار شبیه سازی نمایند.

✓ **Oscillator:** می‌دانیم

که پردازنده‌ها از تعداد زیادی مدارهای تریبی و ترکیبی ساخته شده‌اند. برای هماهنگی بین قسمت‌های مختلف نیاز به یک پالس ساعت داریم که همه قسمت‌ها را با هم هماهنگ کند. همان‌طور که در شکل ۷ دیده می‌شود، برای تهیه این پالس چهار روش وجود دارد:



شکل ۷

الف) اسیلاتور خارجی

ب) اسیلاتور RC داخلی، (در هنگام ساخت، میکروکنترلر را روی فرکانس داخلی 1MHz تنظیم می‌کنند)

ج) اسیلاتور RC خارجی $f = 1/RC$

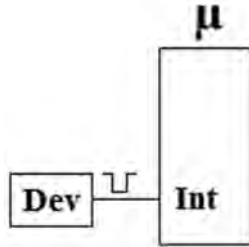
د) اسیلاتور داخلی با کریستال خارجی

وقفه (Interrupt): برای سرویس دهی به هر دستگاه جانبی دو روش وجود دارد:

سرکشی Polling

وقفه Interrupt

در روش سرکشی، CPU موظف است در فاصله‌های زمانی مشخص به دستگاه جانبی سرکشی کند، تا در صورت نیاز به آن دستگاه سرویس لازم را ارائه نماید. در این روش وقت زیادی از CPU تلف می‌شود. اما در روش وقفه دستگاه هر موقع نیاز داشت، با ارسال سیگنال وقفه درخواست سرویس می‌کند و میکرو کار خود را قطع کرده و به آن دستگاه سرویس می‌دهد، در نتیجه زمان تلف نمی‌شود. در هر میکرو چند وقفه خارجی وجود دارد، شکل ۸. در میکرو Mega8 پایه‌های Int^0 و Int^1 مربوط به وقفه‌های خارجی هستند.



شکل ۸

Sleep: برای ساخت دستگاه‌های قابل حمل (پرتابل) لازم است که تا حد امکان توان کمتری مصرف شود تا طول عمر باتری بیشتر شود. بنابراین لازم است که میکرو بتواند در هنگام بیکاری به حالت خواب (Sleep) برود تا توان مصرفی میکرو کاهش پیدا کند. در این میکروکنترلر شش روش Sleep پیش بینی شده است.

■ مراحل ایجاد پروژه با استفاده از ویزارد (Wizard):

۱ از منوی Tools یا از نوار ابزار، CodeWizardAVR را انتخاب کنید، شکل ۹.



شکل ۹



شکل ۱۰

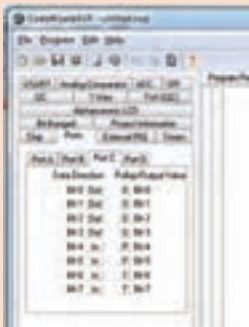
۲ در پنجره CodeWizardAVR که باز می‌شود، گزینه اول که در آن ATmega وجود دارد، به طور پیش‌فرض انتخاب شده است. با کلیک بر روی OK به مرحله بعد می‌رویم، شکل ۱۰.



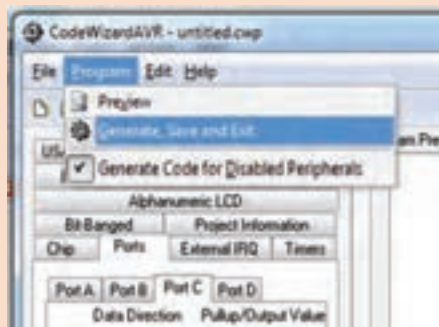
شکل ۱۱

۳ پنجره Wizard در حالی که برگه Chip انتخاب شده است، باز می‌شود. در این برگه شماره میکروکنترلر و فرکانس کاری آن را مشخص می‌کنیم، شکل ۱۱.
۴ با توجه به نیاز پروژه، برگه‌های دیگر را باز و تنظیم‌های لازم را انجام می‌دهیم.

در برگه **Ports**، اگر پورتی به عنوان ورودی تنظیم شده و در قسمت **pull-up** حرف **T** انتخاب شود، یعنی این پایه **HI-Z** است و منطق لاجیکی آن بستگی به ورودی دارد. اگر حرف **T** را با کلیک به **P** تبدیل کنید، این پایه از داخل میکرو **pull-up** شده و منطق یک خواهد داشت، شکل ۱۲.



شکل ۱۳



شکل ۱۲

۵ پس از اتمام تنظیمات، از مسیر «Program/Generate, Save and Exit» پروژه و فایل‌های مربوط به آن را با نام مناسب و در محلی مشخص ذخیره می‌کنیم، شکل ۱۳.

در این مرحله لازم است سه فایل **Source**، **Project** و **CWP** را ذخیره کنیم، شکل ۱۴.

نکته



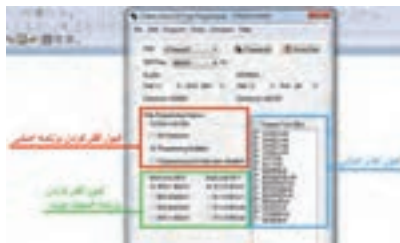
نکته





شکل ۱۴

پس از ذخیره فایل‌ها، کد برنامه با توجه به تنظیمات ویزارد باز می‌شود. حال می‌توان برنامه مورد نظر را به آن اضافه کرد.



شکل ۱۵

فیوز بیت (Fuse bit): فیوز بیت‌ها تعدادی بیت در حافظه Flash هستند که می‌توان آنها را توسط پروگرامر خواند، ویرایش کرد و دوباره برنامه‌ریزی نمود. در شکل ۱۵ فیوز بیت‌های Mega32 نشان داده شده است.

هر دسته یا هر یک از فیوز بیت‌ها می‌توانند بخشی از میکرو را در وضعیت مشخصی قرار دهند.

۳-۲-۱- کاربرد بعضی از فیوز بیت‌های اصلی به شرح زیر است:

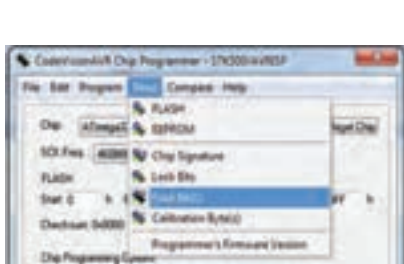
- ✓ **CKSEL:** توسط این چهار فیوز می‌توان مشخص کرد که اولاً منبع پالس ساعت (کلاک) سیستم از کجا تأمین شود و ثانیاً فرکانس آن چقدر باشد.
- ✓ **BODEN:** برای فعال شدن Brown-Out Detection باید این فیوز فعال شود.
- ✓ **BODLEVEL:** این فیوز بیت مشخص می‌کند Brown-Out Detection روی چه ولتاژی عمل کند:

۱ = ۲/۷ Volt

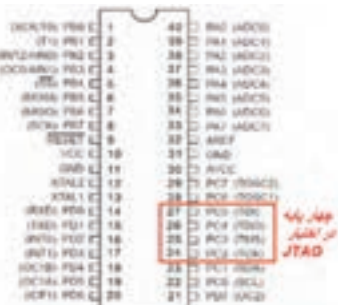
۰ = ۴ Volt

✓ **EESAVE**: اگر این فیوز بیت فعال باشد هنگام پاک کردن حافظه، Flash پاک می‌شود ولی حافظه دیتای ماندگار EEPROM پاک نمی‌شود.

✓ **JTAGEN**: برای فعال شدن JTAG باید این فیوز فعال باشد. توجه داشته باشید وقتی از میکروکنترلر برای اولین بار استفاده می‌کنید، این فیوز فعال است. در نتیجه چهار پین از PORTC در اختیار JTAG است، شکل ۱۶ و در نتیجه نمی‌توانید از تمام پایه‌های پورت C استفاده کنید.



شکل ۱۷



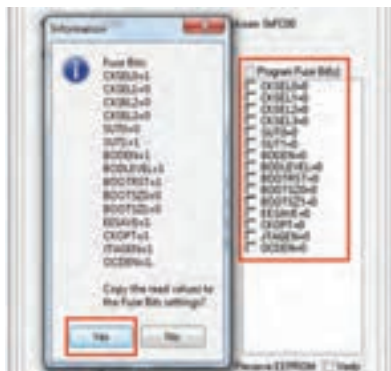
شکل ۱۶

✓ **OCDEN (On-Chip Debug Enabled)**: اگر این فیوز، همراه با فیوز JTAGEN فعال باشد، آنگاه می‌توانید با استفاده از رابط JTAG، برنامه داخل میکرو را خط به خط اجرا کرده و در صورت نیاز رفع اشکال نمایید.

✓ **خواندن فیوز بیت‌ها**: برای این منظور در پنجره Chip Programmer مسیر نشان داده شده در شکل ۱۷ را طی می‌کنیم. بعد از این مرحله، فیوزها خوانده شده و در یک پنجره مانند شکل ۱۸ نمایش داده می‌شود. اگر گزینه **yes** را انتخاب کنید وضعیت موجود فیوزها، به قسمت Program Fuse Bit(s) منتقل خواهد شد و امکان ویرایش آنها وجود دارد.



شکل ۱۹



شکل ۱۸

✓ برنامه‌ریزی فیوز بیت‌ها: برای این منظور مانند شکل ۱۹ در پنجره Chip Programmer مسیر زیر را طی می‌کنیم.

نکته

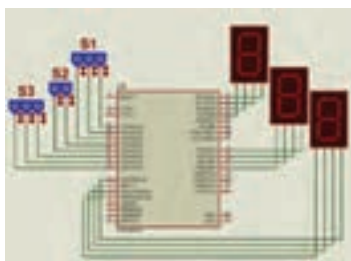


۱- توجه داشته باشید که ابتدا با مراجعه به برگه اطلاعاتی (Data sheet) میکروکنترلر اطلاع کاملی از وضعیت Fuse Bitها پیدا کنید و سپس مبادرت به تغییر آنها نمایید. در غیر این صورت ممکن است میکرو در حالتی قرار گیرد که دیگر در مدار شما کار نکند و نتوانید با پروگرامرهای معمولی آن را از این حالت خارج کنید.

۲- در صورتی که هنگام پروگرام کردن میکرو، تیک مربوط به program fuse bit(s) فعال باشد، ممکن است فیوز بیت‌های CKSEL بر روی کریستال خارجی یا حالت دیگری قرار بگیرد. در این صورت لازم است که یک موج مربعی با فرکانس ۱MHz به پایه XTAL۱ متصل کنید تا بتوانید دوباره فیوز بیت‌ها را برنامه‌ریزی کنید.

■ کاربرد عملگر shift و AND و OR:

مثال ۱: فرض کنید مانند شکل ۲۰ سه عدد سنسور که خروجی آنها به فرم دیجیتال است به PORTA متصل باشد. می‌خواهیم عدد هر سنسور را روی پورت‌های A، B و C نمایش دهیم. در جدول ۲۳ برنامه مربوط به این مثال نوشته شده است.



شکل ۲۰

جدول ۲۳- برنامه مثال ۱

<pre>#include <mega32.h> unsigned char a,b,c,d; void main (void) { DDRA=0x00; DDRB=0xff; DDRC=0xff; DDRD=0xff; while(1) { </pre> <p>ادامه در ستون مقابل</p>	<pre>a=PINA; b=a &(0b00000111); c=(a &(0b00011000))>>3; d=(a &(0b11100000))>>5; PORTB=b; PORTC=c; PORTD=d; } }</pre>
---	--

■ آرایه (ARRAY):

اگر به تعداد زیادی متغیر از یک نوع نیاز باشد، بهتر است برای معرفی آنها از آرایه یا متغیرهای اندیس‌دار استفاده کنیم که به شکل زیر تعریف می‌شوند:

{ , , مقدار دوم , مقدار اول } = [تعداد خانه‌ها] نام آرایه نوع متغیر

int d[5] = { 7 , 12 , 0 , 99 , 20 };

d[0] d[1] d[2] d[3] d[4]

7	12	0	99	20
---	----	---	----	----

d[4] = d[0] + 5;

d[3]++;

d[2] = d[1] * 5;

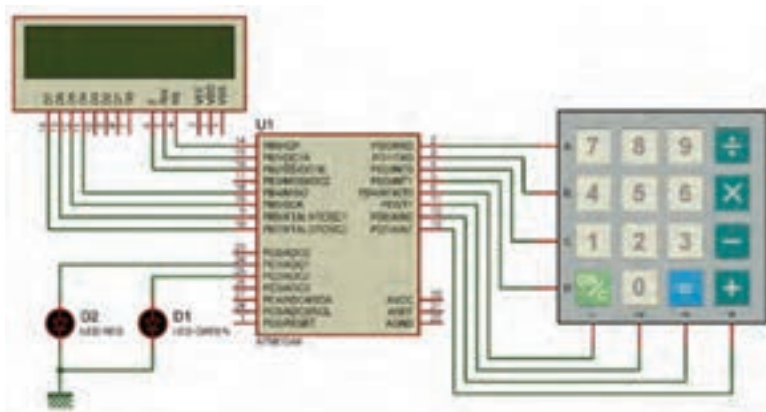
d[1]--;

d[0] = 8;

جدول ۲۴- مقادیر خانه‌های آرایه قبل و بعد از انجام عملگرها

آرایه	d[0]	d[1]	d[2]	d[3]	d[4]
مقدار	7	12	0	99	20
بعد از عملیات	8	11	60	100	12

مثال ۲: با توجه به شکل ۲۱ برنامه‌ای بنویسید که کاربر یک رمز چهار رقمی را وارد کند در صورتی که رمز صحیح بود، LED۱ و در صورت غلط بودن، LED۲ روشن شود و بعد از یک ثانیه خاموش شود. در جدول ۲۵، برنامه با فرض این که رمز ۱۳۹۷ باشد نوشته شده است.



شکل ۲۱

در برنامه بالا هر کلیدی که زده شود در یکی از خانه‌های آرایه d ذخیره می‌شود. متغیر i مشخص می‌کند که عدد کلید زده شده در کدام خانه آرایه ذخیره شود. با هر کلیدی که ذخیره می‌شود، متغیر i یک واحد افزایش می‌یابد. هر گاه $i=4$ شود یعنی یک رمز چهار رقمی وارد شده که مورد ارزیابی قرار می‌گیرد و با توجه به غلط یا صحیح بودن عمل مورد نظر انجام می‌شود.

جدول ۲۵- پاسخ مثال ۲

<pre>#include <mega8.h> #include <delay.h> #include <stdio.h> #include <alcd.h> #define R1 PORTD.0 #define R2 PORTD.1 #define R3 PORTD.2 #define R4 PORTD.3 #define C1 PIND.4 #define C2 PIND.5 #define C3 PIND.6 #define C4 PIND.7 unsigned char kb(void); unsigned char key,i=0,d[20]; char s[4]; void main (void) { DDRC=0x06; DDRD=0x0f; lcd_init(16); lcd_putsf("Enter Pass:"); while(1) { key=kb(); if (key !=16) { sprintf (s,"%d",key); lcd_puts(s); d[i]=key; i++; if(i==4) { if(d[0]==1&&d[1]==3&&d[2]= =9&&d[3]==7) PORTC.2=1; else PORTC.1=1; delay_ms(1000); PORTC=0x00; i=0; d[0]=d[1]=d[2]=d[3]=0; lcd_clear(); lcd_putsf("Enter Pass:"); } } } }</pre> <p>ادامه برنامه در ستون مقابل از بالا به پایین</p>	<pre>//----- kb function ----- unsigned char kb (void) { unsigned char k=16; PORTD=0xFF; //-----ROW1----- R1=0; delay_ms(3); if (C1==0) {k=7 ; while (C1==0);} if (C2==0) {k=8 ; while (C2==0);} if (C3==0) {k=9 ; while (C3==0);} if (C4==0) {k=10 ; while (C4==0);} R1=1; //-----ROW2----- R2=0; delay_ms(3); if (C1==0) {k=4 ; while (C1==0);} if (C2==0) {k=5 ; while (C2==0);} if (C3==0) {k=6 ; while (C3==0);} if (C4==0) {k=11 ; while (C4==0);} R2=1; //-----ROW3----- R3=0; delay_ms(3); if (C1==0) {k=1 ; while (C1==0);} if (C2==0) {k=2 ; while (C2==0);} if (C3==0) {k=3 ; while (C3==0);} if (C4==0) {k=12 ; while (C4==0);} R3=1; //-----ROW4----- R4=0; delay_ms(3); if (C1==0) {k=15 ; while (C1==0);} if (C2==0) {k=0 ; while (C2==0);} if (C3==0) {k=14 ; while (C3==0);} if (C4==0) {k=13 ; while (C4==0);} R4=1; return k; }</pre>
--	--

اگر می‌خواهید هنگام ورود رمز، عدد ورودی دیده نشود و به جای آن کاراکتر * چاپ شود، مانند جدول ۲۶ دو خط زیر را با کد داده شده عوض کنید.

جدول ۲۶ - تغییر دستور برای چاپ *

کد اول	کد دوم
<pre>sprintf(s,"%d",key); lcd_puts(s);</pre>	<pre>lcd_putchar('*');</pre>

■ تنظیمات کدویژن برای پروگرامر و نحوه پروگرام کردن:

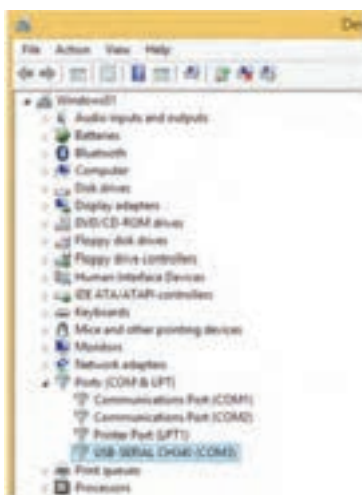
پس از نصب و شناخته شدن پروگرامر توسط ویندوز، برنامه کد ویژن را اجرا کرده و مانند شکل ۲۲ از منوی Setting گزینه Programmer را انتخاب کنید.



شکل ۲۳



شکل ۲۲

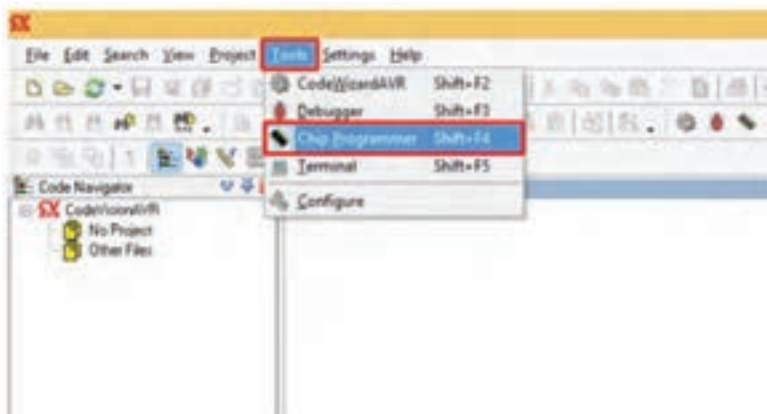


شکل ۲۴

در پنجره باز شده موسوم به Programmer Setting، نوع پروگرامر و پورتی که پروگرامر به آن متصل است را مشخص کنید، شکل ۲۳. در این شکل، تنظیمات با فرض این که نوع پروگرامر / Atmel STK500 AVRISP و AVRISP و پورت متصل شده COM3 باشد، صورت گرفته است.

شماره پورتی که پروگرامر به آن متصل است را می‌توانید مانند شکل ۲۴ در قسمت Device Manager مشاهده کنید.

■ نحوه پروگرام کردن میکروکنترلر در محیط کدویژن:
ابتدا مانند شکل ۲۵ از منوی Tools گزینه Chip Programmer را انتخاب کنید.



شکل ۲۵

حال در پنجره Chip Programmer می‌توانید پس از انجام تنظیم‌های لازم با کلیک بر روی دکمه Program All برنامه را به میکرو منتقل کنید، شکل ۲۶.



شکل ۲۶

توجه: بهتر است تیک قسمت Program Fuse Bit برداشته شود تا فیوز بیت‌ها به‌طور ناخواسته در حالتی قرار نگیرند که میکرو در مدار شما قادر به کار نباشد.

بیشترین مشکل مربوط به فیوز بیت‌های CKSEL است. وقتی که منبع پالس بر روی پالس خارجی یا RC خارجی قرار گیرند ولی این دو منبع به میکروکنترلر وصل نباشند، پالس مورد نیاز میکرو تأمین نمی‌شود و میکرو کار نخواهد کرد.

■ آردوینو (Arduino):

آردوینو یک پلتفرم (ساختار یا بستر) متن‌باز (Open Source Platform) است که افراد مختلف به کمک آن می‌توانند انواع پروژه‌های الکترونیکی را به راحتی بر روی آن

انجام دهند. اصطلاح «پلتفرم متن‌باز» به این معنا است که نرم‌افزار و سخت‌افزار آن به صورت آزاد در اختیار عموم قرار می‌گیرد. این امکان وجود دارد که اشخاص خود اقدام به ساخت برد آردوینو برای انجام پروژه‌های خود کنند. با این حال انواع مختلفی از بردهای آردوینو به صورت آماده برای کاربری‌های گوناگون وجود دارند که می‌توان آنها را از بازار تهیه نمود. متن‌باز بودن این پلتفرم باعث می‌شود که میلیون‌ها نفر در سرتاسر جهان اقدام به توسعه هر چه بیشتر سخت‌افزار و نرم‌افزار آن کنند. از مزیت‌های آردوینو این است که هر کس می‌تواند یک مدار مکمل که شیلد (Shield) نامیده می‌شود، برای کاری خاص برای آن طراحی کند و علاوه بر آن کتابخانه‌ای نیز برای راحت‌تر کار کردن با آن شیلد، توسط طراح آن توسعه می‌یابد و اغلب به صورت رایگان در اختیار عموم نیز قرار می‌گیرد. در نتیجه تعداد بسیار زیادی مثال و کتابخانه آماده برای آردوینو موجود است.

☑ معرفی سخت‌افزار آردوینو:

آردوینو یک میکروکنترلر تک‌بردی است که به منظور تولید راحت‌تر برنامه‌هایی که با اشیا یا محیط تعامل داشته باشند، طراحی شده است. سخت‌افزار آن به صورت متن‌باز در اختیار عموم قرار گرفته است که بر مبنای یک میکروکنترلر Atmel طراحی شده است. به عنوان مثال، مدل Arduino UNO Rev3 دارای یک رابط USB، شش پین ورودی آنالوگ و همچنین چهارده پین ورودی/خروجی دیجیتال است که اجازه اتصال بردهای توسعه مختلفی را فراهم می‌آورد. آردوینو مدل‌های مختلفی دارد که در آنها تعداد ورودی‌های آنالوگ و پین‌های دیجیتال ممکن است بسته به میکروکنترلر استفاده شده، بر روی آنها کم و یا زیادتر باشد. آردوینو می‌تواند جهت ایجاد اشیای تعاملی، گرفتن ورودی از تعداد زیادی سنسور و کنترل ادوات الکترونیکی گوناگون از طریق خروجی‌ها به کار گرفته شود. آردوینو به همراه یک IDE عرضه شده است که می‌توان به کمک آن، با زبان‌های C و C++ اقدام به برنامه‌نویسی نمود، شکل ۲۷. IDE مربوطه را می‌توان از سایت www.arduino.cc به صورت رایگان دریافت کرد.


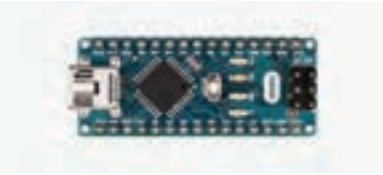




شکل ۲۷

آردوینو در اینترنت اشیا کاربرد بسیاری دارد. به عنوان مثال آردوینو می‌تواند پارامترهایی مانند نور محیط، کلیدها و حتی یک ایمیل را به عنوان ورودی دریافت نماید و بعد از پردازش، خروجی‌هایی مانند روشن کردن یک وسیله برقی، تغییر رنگ LEDها یا ارسال یک ایمیل یا نظیر آن را ارائه دهد.

بردهای آردوینو دارای یک آی سی مبدل USB به سریال است که به وسیله آن به سادگی و بدون نیاز به پروگرامر می‌توان آنها را برنامه‌ریزی کرد. این کار به سبب وجود Bootloader مخصوص در میکروکنترلر استفاده شده در آردوینو، امکان‌پذیر شده است. همچنین از طریق این مبدل می‌توان اقدام به رفع اشکال کردن از برنامه نوشته شده نمود. به عنوان مثال می‌توان خروجی‌های چند سنسور را به صورت نمودار بر روی رایانه مشاهده کرد. در اکثر بردهای آردوینو از میکروکنترلر Atmega328p استفاده شده است. برای پیدا کردن مثال‌ها و کتابخانه‌های گوناگون می‌توان به سایت github.com مراجعه نمود.

جدول ۲۷- چند نمونه از بردهای Arduino

	<p>Arduino Uno</p>
	<p>Arduino Nano</p>
	<p>Arduino Mega</p>
	<p>Arduino DUE</p>

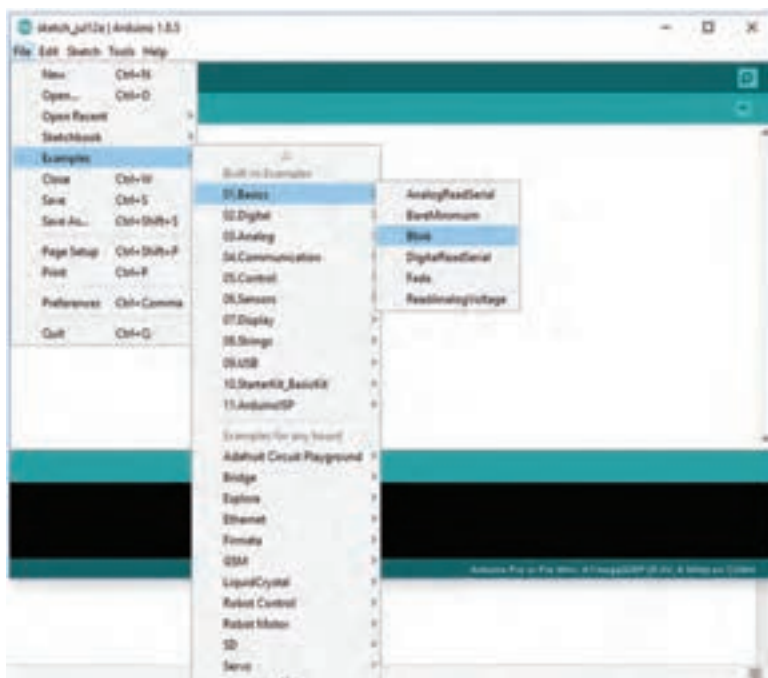
✓ مشخصات برد **Arduino Uno** :

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
LED_BUILTIN	13
Length	68.6 mm
Width	53.4 mm
Weight	25 g

✓ **برنامه‌نویسی در محیط Arduino**: بعد از نصب و اجرای **Arduino IDE** از منوی بالایی به گزینه **File** رفته و **New** را انتخاب می‌کنیم. یک سند جدید برای برنامه‌نویسی ایجاد می‌شود. در داخل این سند دو تابع اصلی برای برنامه‌نویسی وجود دارد.

ابتدا به معرفی تابع `void Setup()` می‌پردازیم، از این تابع برای پیکربندی و ایجاد تنظیمات اولیه استفاده می‌شود. به عنوان مثال، تنظیمات پین‌های ورودی و خروجی، پورت سریال، I_{DC} و دیگر مشخصات باید در این بخش انجام شود. تابع `void loop ()`

شامل برنامه اصلی است و این تابع به صورت مکرر اجرا می شود.
☑ مثال های آماده برای آردوینو: آردوینو دارای بخش Examples شکل ۲۸ است.
در این بخش می توان به مجموعه ای از مثال های ساده موجود در IDE دسترسی پیدا کرد.
به عنوان مثال برنامه چشمک زن را اجرا می کنیم.



شکل ۲۸

از منوی فایل Examples را انتخاب کرده و از زیر شاخه Basics گزینه Blink را انتخاب می‌کنیم.

```
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

در برنامه بالا در قسمت Void setup() ابتدا پین مربوط به LED را به‌عنوان خروجی در نظر می‌گیریم. این کار با دستور pinMode انجام می‌شود. بر روی بردهای Arduino یک LED از پیش تعیین شده وجود دارد که بر روی پین ۱۳ قرار گرفته است. برای تعیین این پین به‌عنوان خروجی، از دستور pinMode(۱۳,OUTPUT) استفاده می‌شود. در قسمت Loop برنامه از طریق دستور digitalWrite مقدار مورد نظر (۰ و ۱) به پین مورد نظر اعمال می‌شود. برای اعمال تأخیر، از دستور Delay استفاده می‌شود. اضافه کردن کتابخانه جدید: معمولاً کتابخانه‌های ارائه شده برای آردوینو به‌صورت فایل Zip می‌باشند. بعد از دانلود کتابخانه مورد نظر، کافی است از منوی Sketch، Include Library رفته و گزینه Add.zip Library را انتخاب کنید.

ارزشیابی مربوط به پودمان دوم

کاربرگ ارزشیابی واحد یادگیری ۳

مرحله کار: طراحی الگوریتم (فلوچارت) مدار پروژه ساده الکترونیکی کار: -	
نام و نام خانوادگی هنرجو:	کد کار:
تاریخ:	تاریخ:
<p>بارم آزمون: ارزشیابی نظری و نرم‌افزاری در نظام ارزشیابی ۲۰ نمره‌ای شامل ۱۵ نمره آزمون و ۵ نمره مستمر است که نمره معادل آن در نظام ارزشیابی مبتنی بر شایستگی ۱+۳ می‌شود.</p> <p>آزمون نظری: سؤال بر اساس الگوی پرسش</p> <p>۱- در فلوچارت‌ها از علامت شروع می‌تواند... فلش خارج و به علامت پایان می‌تواند... فلش وارد شود؟</p> <p>الف) یک - یک (ب) یک - چند (پ) چند - یک (ت) چند - یک</p> <p>۲- برای الگوریتم زیر یک فلوچارت رسم کنید.</p>	
<p>۱- شروع</p> <p>۲- اعداد صحیح و مثبت a و b را بگیر</p> <p>۳- $1 \leftarrow c$</p> <p>۴- اگر $b=0$ است، به مرحله ۸ برو</p> <p>۵- $c \leftarrow c.a$</p>	<p>۶- $1 \leftarrow b$</p> <p>۷- اگر $b \neq 0$ است، به مرحله ۵ برو</p> <p>۸- c را چاپ کن</p> <p>۹- پایان</p>
<p>۳- محدوده متغیرهای <code>unsigned char</code> و <code>int</code> را با محاسبه مشخص کنید؟</p> <p>۴- کدام یک از تعاریف زیر صحیح است؟</p> <p>الف) <code>int i=۳۷</code>; (ب) <code>char x='A'</code>; (پ) <code>float if=۱۲/۷</code>; (ت) <code>unsigned char i=۳۰۰</code>;</p> <p>۵- اگر $a=0x57$ و $b=0xc9$ باشد حاصل عبارت $(a \ll 3) \wedge (a \& b)$ چه عددی خواهد شد.</p> <p>۶- در زبان C معرفی توابع..... انجام می‌شود؟</p> <p>الف) قبل از متغیرها (ب) بعد از <code>main</code> (پ) قبل از <code>main</code> (ت) در داخل برنامه</p>	
<p>آزمون نرم‌افزاری: سؤال بر اساس الگوی پرسش -</p>	
<p>آزمون سخت‌افزاری: سؤال بر اساس الگوی پرسش</p>	
<p>بارم آزمون: ۲۰ نمره</p>	
<p>شایستگی‌های غیر فنی: مشابه ارزشیابی مرحله اول از واحد یادگیری (کار) شماره ۱</p>	
<p>کلیه آزمون‌ها بر اساس استاندارد عملکرد نمونه برگ ۸-۱ انجام می‌شود.</p>	

کاربرگ ارزشیابی واحد یادگیری ۴

مرحله کار: برنامه‌نویسی به زبان C و تحلیل برنامه‌های آماده پروژه‌های الکترونیکی کار:		
نام و نام خانوادگی هنرجو:	کد کار:	تاریخ:
<p>بارم آزمون: ارزشیابی نظری و نرم‌افزاری در نظام ارزشیابی ۲۰ نمره‌ای شامل ۱۵ نمره آزمون و ۵ نمره مستمر است که نمره معادل آن در نظام ارزشیابی مبتنی بر شایستگی ۱+۳ می‌شود.</p> <p>آزمون نظری: سؤال بر اساس الگوی پرسش</p> <p>۱- در حلقه $for(i=7; i < 16; i=i+2)$ دستور $PORTA=1$ چند بار اجرا می‌شود؟ الف) ۹ ب) ۸ پ) ۶ ت) ۵</p> <p>۲- برنامه‌ای بنویسید که با گرفتن یک عدد از روی پورت A، led موجود روی پورت B را به تعداد عدد خوانده شده روشن و خاموش کند.</p> <p>۳- با اجرای برنامه زیر چه اعدادی روی PORTD دیده خواهد شد.</p> <pre>for(i=0; i<10; i++) {PORTD=i*2; delay_ms(500);}</pre> <p>۴- اگر به یک پورت LED متصل باشد، لازم است پایه‌های پورت Pull-Up داخلی شوند؟ صحیح <input type="checkbox"/> غلط <input type="checkbox"/></p> <p>۵- اگر به پورت B میکروکنترلر، LED متصل باشد، دستور $DDRB=0x55$ باعث می‌شود. الف) LEDها یکی در میان روشن شود ب) LEDها کامل روشن شود پ) پورت B یکی در میان خروجی شود ت) پورت B کامل خروجی شود</p> <p>۶- اگر هشت عدد LED به پورت B متصل باشد، با اجرای برنامه زیر آنها به چه ترتیبی روشن می‌شوند با استفاده از ۰ و ۱ پاسخ دهید. (۰ برای خاموش و ۱ برای روشن)</p> <pre>While(1) { for(i=0; i<8; i++) { PORTB=1<<i; delay_ms(500); } }</pre> <p>۷- اگر مقدار دو سنسور دما $sa=25$ و $sb=33$ باشد، با توجه به دستور زیر آیا fan روشن می‌شود؟ <pre>if (a>22&& b>30) fan=1;</pre></p> <p>۸- با فرض این که رمز یک مدار قفل رمز ۱۳۹۷ باشد، با استفاده از دستور شرطی <code>if _ else</code> و عملگر <code>and</code> قطعه برنامه‌ای بنویسید که اگر رمز صحیح بود، بیت ششم PORTD و اگر اشتباه بود، بیت هفتم PORTD یک شود.</p> <p>۹- ۱- اگر ولتاژ مرجع $2/56V$ و تعداد بیت خروجی مبدل ADC ۱۰ بیت باشد و یک سنسور دما با حساسیت 10 mv/C به کانال سوم وصل باشد، برنامه‌ای بنویسید که بیت پنجم PORTD در دمای بالاتر از 30°C، برابر یک و در دمای کمتر از 25°C، برابر صفر شود.</p>		
آزمون نرم‌افزاری: سؤال بر اساس الگوی پرسش -		
آزمون سخت‌افزاری: سؤال بر اساس الگوی پرسش		
شایستگی‌های غیر فنی: مشابه ارزشیابی مرحله اول از واحد یادگیری (کار) شماره ۱		
کلید آزمون‌ها بر اساس استاندارد عملکرد نمونه برگ ۸-۱ انجام می‌شود.		