مطالبی که در این شماره میخوانیم :

....

شماره -۲

Persian Microcontroller Magazine

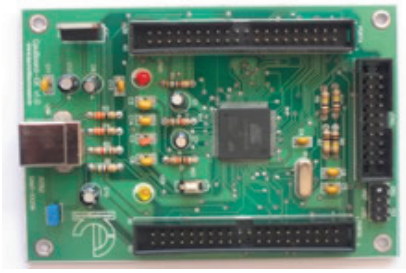
PMM

شماره ۲

کویر الکترونیک شهریور ۱۳۸۸

Welcome to the world of Atmel' ARM-based Microcontrollers

ARM POWERED



Evaluation Board SAM7X

- Low power, high performance, scalable architecture
- Wide portfolio to cover all ranges
- Rapid time to market using readily available software
- Widest range of hardware and software tools support
- Widest range of RTOS and software components

سلا م

با یاری خداوند متعال دومین شماره مجله PMM منتشر شد .

همانگونه که در شماره قبل قول دادیم ،کل مطالب این شماره را به میکرو کنترلر های ARM اختصاص داده ایم .

در این مجله ، کلیه کار های که تاکنون در مبحث ARM انجام شده در اختیار شما قرار میگیرد ، شما بعد از خواندن این شماره قادر به نوشتن برنامه های ساده همچون چشمک زن و... خواهید .

مطالب به صورت زنجیر وار اورده شده شده است و هر مبحث به سوالاتی که در ذهن شما وجود دارد پاسخ خواهد داد .در صورتی که پاسخ خود را نیافتید در لینک های که در انتها اورده شده است مطرح کنید ، دوستان شما در اسرع وقت به شما پاسخ خواهند داد .

با تشکر ویژه از مدیریت سایت IR-MAN.COM

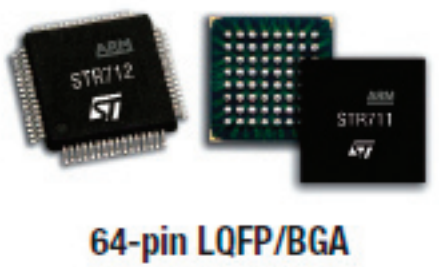
گروه کویر الکترونیک

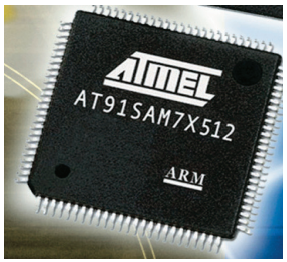
arm چیست ، چرا arm

arm چیست و چه فرقی با xmega دارد؟arm بهتر است یا avr ۳۲ ؟چرا باید از arm استفاده کنیم ؟ایا arm یک میکرو کنترلر صنعتی است ؟فرق arm با میکرو های pic چیست ؟,.....

علی رغم اینکه در اکثر سایت ها و مطالب اموزشی ارائه شده از طرف دوستان به این سوال پاسخ داده شده است، باز هم عده ای از دوستان سوالات را در قالب های مختلف تکرار می کنند ، در ادامه به مقایسه ای اجمالی در مورد این میکرو کنترلر ها پرداخته ایم تا به تمامی سوالات شما پاسخ داده باشیم ، ما قابلیت های arm را تحلیل و بررسی کرده ایم تا شما بتوانید این ویزگی ها را با سایر میکرو کنترلر ها مقایسه کنید و در اخر بهترین میکرو را انتخاب نمایید .

در صورتی که قصد خرید یک میکرو کنترلر قدرمند را داشته باشید ، فروشنده موارد زیر را به شما پیشنهاد میکند :





خانواده Avr

خانواده Pic

خانواده ARM

خانواده Fpga

و سایر میکرو کنترلر ها .

هر کدام از خانواده های بالا خود دارای زیر مجموعه های بسیاری میباشند،در جدول زیر به مقایسه ای کلی میان این ٤ خانواده پرداخته ایم :

سری	تعداد زیر مجموعه	حداکثر فرکانس کاری	منابع یادگیری	قیمت	قدرت پردازش(عمومی)	قدرت پردازش (اختصاصی)	نویز پذیری	پشتیبانی از پروتکل های ارتباطی
avr خانواده	بیش از ۱۲۰	۳۰۰ مگا هرتز	خیلی زیاد	نسبتا ارزان	متو سط	ضعیف	زیاد	متو سط
pic خانواده	بیش از ٦۰	٤۰ مگا هرتز	زیاد	متو سط	متو سط	متو صط	کم	خوب
arm خانواده	بیش از ۲۰۰	بیش از ۱گیگا هرتز	متو سط	متو سط	بالا	بالا	کم	خیلی خوب
fpga خانواده	بیش از ۲۰۰	بیش از ۱گیگا هرتز	متو سط	متو سط	متو سط	بالا	کم	متو سط

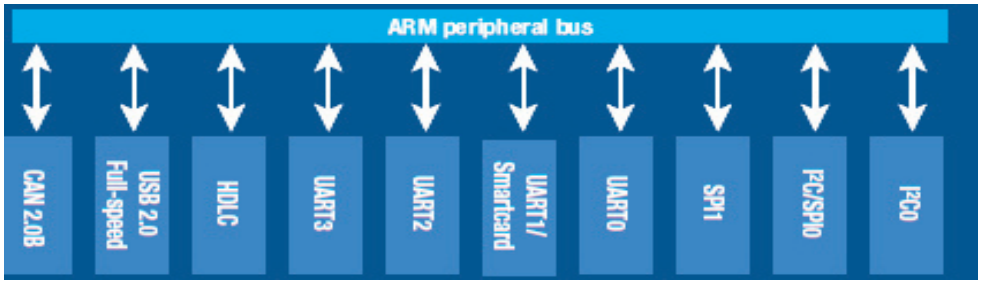
منظور از قدرت پردازش عمومی و اختصاصی ، سرعت و قدرت پردازش اطلاعات در مصارف عمومی (مانند کار های کنترلی ، ...) و اختصاصی (مانند پردازش تصویر و...) میباشد .

هر یک از خانواده های بالا خود دارای دسته بندی های می باشند مثل خانواهد avr خود به دسته های avr۳۲ ، xmega ، mega ، tiny و... تقسیم میشود و هر دسته خود دارای میکرو کنترلر های مخصوص به خود هست.

یکی از مواردی که عموما در انتخاب میکرو مورد توجه قرار میگیرد ، پشتیبانی میکرو از پروتکل های ارتباطی ارتباطی است . در این میان میکرو arm از بیشترین پروتکل های موجود پشتیبانی میکند .

پروتکل های که عموما در ایران استفاده میشود عبارتند از usb ، rs۲۳۲ ، spi ، i۲c ، can و... میکرو کنترلر های خانواده arm از این پروتکل ها + چند پروتکل دیگر که در ادامه توضیح داده میشود پشتیبانی میکنند و تمامی انها را به صورت یکجا در خود دارند (مثلا at۹۱sam۷x۲۵٦ ، از تمامی این پروتکل پشتیبانی میکند اما avr ها فقط از چهار مورد) .

و....



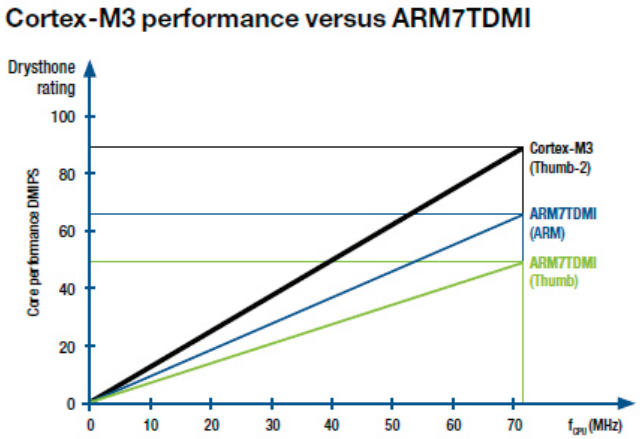
و در نهایت معیار اصلی انتخاب میکرو نیاز شما میباشد . بدون شک بحث بر سر مقایسه میکروکنترلر ها هیچ گاه به پایان نمیرسد و این شما هستید که باید با توجه به نیاز خود بهترین گزینه را انتخاب کنید .

مثلا من به دنبال پردازش تصویر و صدا هستم ، من میتوانم از تمامی میکرو کنترلر های ذکر شده برای کار خود استفاده کنم اما عاقلانه ترین راه استفاده از avr۳۲ یا تراشه های fpga هست ، زیرا در این میکرو کنترلر ها امکاناتی برای کار با صدا و تصویر تعبیه شده است . یا فرد دیگری قصد ساختن یک سیستم کنترلی را دارد . این فرد میتواند از میکرو کنترلر های pic یا arm استفاده کند ، در این حالت میکرو کنترلر های arm امکانات بیشتری را در اختیار او میگزارند .این شخص میتواند با استفاده از پروتکل اترنت (که در اکثر میکرو کنترلر های arm وجود دارد)، اطلاعات سیستم را به مراکز دور ارسال کند .

شما میتوانید اطلاعات جامعی را در مورد میکرو کنترلر های avr و pic و تراشه های fpga در سطح اینترنت بیابید و انها را با امکانات arm که در زیر بررسی شده است مقایسه کنید .

arm یک معماری است که توسط شرکت arm طراحی شده است ، شرکت های مختلف این معماری که همیشه در حال پیشرفت میباشد را خریداری کرده و به ان امکانات جانبی همچون پورت ، حافظه ، پروتکل های مختلف و... را اضافه میکند . این امر سبب میشود که ما از یک هسته محصولات مختلفی را به وجود اوریم . هسته های که تاکنون ارائه شده اند arm۱۱ , arm۹ , arm۷ میباشند . همچنین از این هسته ها زیر مجموعه های همچون Cortex-M۳ و ... بوجود امد ، خانواده های بالا نیز دچار تغییر تحول توسط عرضه کننده های میکرو گردید و اموزه میبینم که arm۷ با نمهای مختلفی عرضه میشود

نحوه برنامه نویسی برای تمامی هسته ها یکسان میباشد و در هر نمونه جدید ، قدرت و سرعت پردازش داده افزایش یافته است .



Arm ها از معماری risc استفاده میکنند .

معماری risc (Reduced Instruction Set Coding/Computer) نوعی طراحی برای سخت افزار های میکرو الکترونیک میباشد.

در این معماری با تغییرات سخت افزاری مجموعه دستورات برنامه نویسی کم میشود،در مقابل این معماری ، معماری cisc (Complex Instruction Set Coding) قرار دارد ، در معماری cisc حجم دستورات زیاد تر میشود ، اما سخت افزار میکرو کنترلر ساده تر میگردد ، مثلا در معماری cisc برای روشن کردن یک led شما باید ، یکی از متغییر های حافظه را تغییر دهید و سپس ان را به cpu ارسال کنید ، cpu بعد از پردازش دستور فرمان set کرد پین را صادر میکند ، اما در معماری risc شما برنامه را در حافظه فلش میکرو قرار میدهید ، بعد از روشن شدن میکرو برنامه به حافظه موقت منتقل میشود و سپس از انجا به سمت cpu میرود ، در cpu دستور پردازش شده و سپس فرمان به کنترل کننده پورت ارسال میشود و.... در نهایت پایه مورد نظر set میگردد .

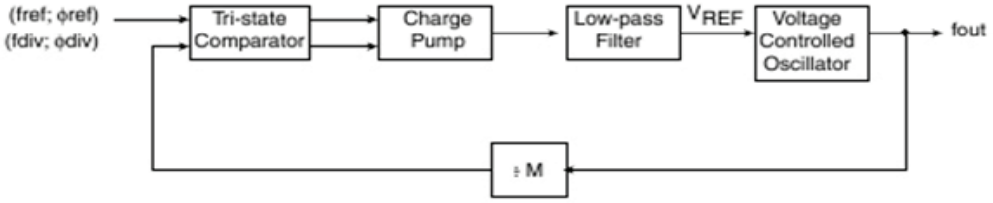
میکرو کنترل های arm دارای حافظه های مختلفی همچون High-speed Flash و sram میباشند .

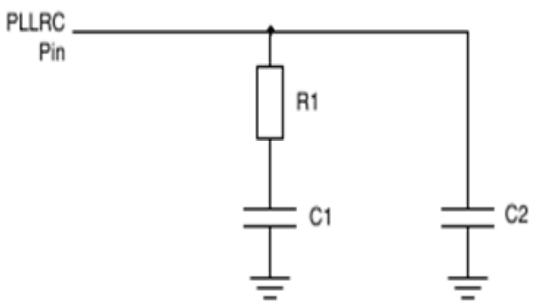
Static random access memory یا SRAM نوعی حافظه موقت میباشد ، که اطلاعات قبل از انکه به CPU برسد در ان ذخیر ه میشود . با این کار دیگر نیاز نیست وقت CPU برای دریافت اطلاعات از حافظه فلش تلف شود ،،در هنگام پردازش اطلاعات توسط CPU اطلاعات از فلش به این حافظه منتقل میشود و همیشه اطلاعات در دسترس CPU خواهد بود . در این نوع از حافظه ها از فلیپ فلاپ برای ذخیره سازی هر بیت داده استفاده می گردد. یک فلیپ فلاپ برای یک سلول حافظه، از چهار تا شش ترانزیستور استفاده می کند . حافظه های SRAM نیازمند بازخوانی / بازنویسی اطلاعات نخواهند بود، بنابراین سرعت این نوع حافظه ، از دیگر حافظه های هم سطح (DRAM و..) بسیار بیشتر میباشد .

همانطور که میدانید برنامه در حافظه Flash میکرو ذخیره میشود (در هنگام پروگرام کردن میکرو برنامه از کامپیوتر به این حافظه منتقل میشود) برای دستیابی به سرعت بالا و عدم تاخیر در خواند حافظه فلش و تلف شدن وقت cpu ، حافظه فلش میکرو کنترلر های arm طوری طراحی میشوند که بتوانند با سرعت بالای ۳۰ مگاهرتز اقدام به تبادل داده با sram و cpu نمایند .

در میکرو کنترلر های arm یک نوسان داخلی نیز تعبیه شده است ، این نوسان ساز میتواند پالس های مورد نیاز میکرو را در فرکانس بالا تامین کند ، زیرا با واحد pll کنترل میشود .

Phase-locked-loop (حلقه فازی قفل شده)یا واحد pll در میکرو کنترلر های arm وظیفه ایجاد یک پالس با فرکانس ثابت را بر عهده دارد . واحد pll برای ایجاد فرکانس نیاز به شبکه RC خارجی دارد . بخش pll در میکرو کنترلر های ساخته شده توسط شرکت اتمل از چهار قسمت زیر تشکیل شده است :





واحد pll برای نوسان سازی به شبکه rc روبرو نیاز دارد مقدار خازن ها و مقاومت از فرمول های خاص ، برای فرکانس های مختلف محاسبه میشوند ، در اموزش برنامه نویسی با نحوه محاسبه انها اشنا خواهیم شد .

این میکروکنترلرها نیزهمانند دیگر میکروکنترلرهای معرفی شده مجهز به تایمر Watchdog میباشد

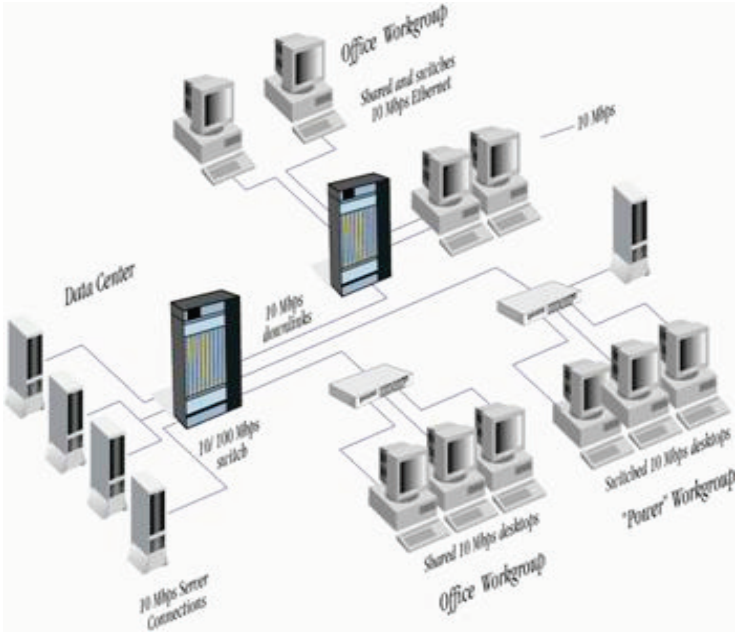
WATCHDOG یکی از تایمر های میکرو است که میتواند تا یک زمان خاص بشمارد وبعد میکرو را ریست کند ، این تایمر میتواند تا ۸ زمان ١٦ ، ٣٢، ٦٤ ، ١٢٨ ، ٢٥٦ ، ٥١٢ ، ١٠٢٤ و ٢٠٤٨ و در بعضی از میکروها ٤٠٩٦, ٨١٩٢ میلی ثانیه بشمارد ، بعد از سپری شدن زمان میکرو ریست میشود و برنامه دوباره از ابتدا اجرا میشود. از این تایمر در زمان وجود خطا یا هنگ کردن میکرو استفاده میشود .

قابلیت که فقط در این خانواده وجود دارد ، پروتکل Ethernet میباشد ، در دیتا شیت این میکرو کنترلر ها ویژگی شبکه به صورت base-T ۱۰/۱۰۰ Ethernet MAC تعریف شده است .

Ethernet :یکی از شبکه های استاندارد کامپیوتری میباشد ، توسط این شبکه شما میتوانید چندین کامپیوتر را به

یکدیگر متصل نمایید و داده بین انها رد و بدل کنید.برای اتصال کامپیوتر به دنیای مجازی نیاز مند یک کارت شبکه هستیم ، میکرو کنتلر های arm میتوانند مستقیما به شبکه Et-ernet متصل شوند و با سایر دستگاه های متصل به شبکه تبادل اطلاعات نمایند .

MAC (Media Access Controller) :منظور از mac قابلیت کنترل دستیابی به فایل ها میباشد ، این پروتکل دسترسی به تمامی لایه های شبکه را برای شما محیا میکند به گونه ای که شما میتوانید به تمام نقاط کامپیوتر یا وسیله متصل شده به شبکه دسترسی داشته باشید (از طریق میکرو خودتان میتواند کنترل کامل وسیله دیگر را در دست بگیرید .)



۱۰/۱۰۰ : این دو عدد بیان گر سرعت شبکه میباشند ، شبکه اترنت در دو سرعت استاندار د ۱۰ Mbit/s و ۱۰۰ Mbit/s در دسترس شما میباشد (البته سرعت بالا تر نیز برای شبکه های کامپیوتری وجود دارد ، اما میکرو کنتلر های arm فقط از این دو سرعت پشتیبانی میکند .)

base-T ((T="Twisted" Pair Copper :شبکه Ethernet درای مبنا های مختلفی میباشد :

شبکه های فیبر نوری

۳.۱ ۱۰۰BASE-FX

شبکه های سیمی

۳.۲ ۱۰۰BASE-SX

۲.۱ ۱۰۰BASE-TX

۳.۳ ۱۰۰BASE-BX

۲.۲ ۱۰۰BASE-T٤

۲.۳ ۱۰۰BASE-T۲

التبه میتوانیم به موارد بالا شبکه های بیسیم را نیز اضافه کنیم .... . اما انچه که باید از base-T بدانیم این است که

این شبکه توسط کابل های زوج تابیده اطلاعات را منتقل میکند . یعنی برای تبادل داده به ۸ سیم نیاز داریم (به غیر از رابط های همزمانی و کلاک) و...

در اموزش برنامه نویسی با Ethernet MAC ۱۰/۱۰۰ base-T بیشتر اشنا خواهیم شد ، شما میتوانید برای دریافت اطلاعات بیشتر به ادرس زیر مراجعه کنید :

http://en.wikipedia.org/wiki/Ethernet

http://en.wikipedia.org/wiki/Fast_Ethernet

یکی دیگر از امکانات ویژه میکرو کنترلر های arm پشتیبانی از پروتکل can میباشد .

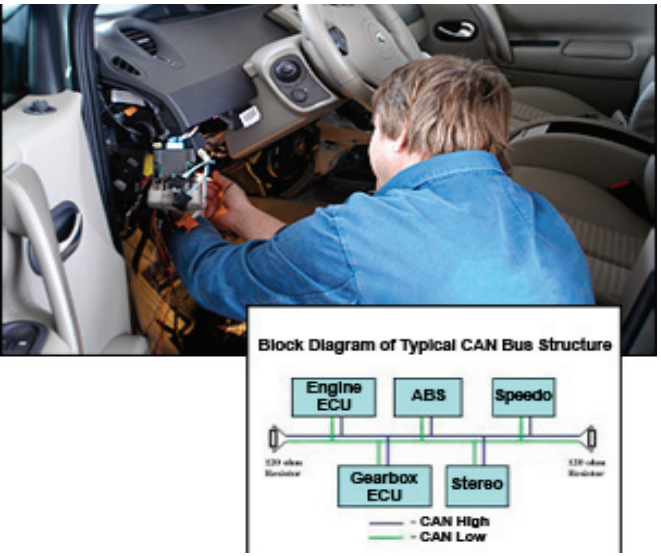
((CAN or CAN-bus Controller-area network یک پروتکل ارتباطی برای متصل کردن میکرو کنترلرها و سایر وسایل الکترونیک به یکدیگر بدون نیاز به وسیله کنترل کننده میباشد .

نمونه قابل مشاهده کاربرد can در ECU اتومبیل میباشد . سیستم ECU تمامی امکانات اتومبیل را کنترل میکند ، امکانی از قبیل ، چراغ ها ، سیستم صوتی ، درب ها و موتور ، سیستم سوخت رسانی و.... قطعا برای اتصال تمامی این امکانات نیاز به تعداد زیادی سیم خواهیم داشت ،

اما با پروتکل can و از طریق دو سیم تمامی موارد را به پردازنه اصلی مرتبط میکنیم و.....

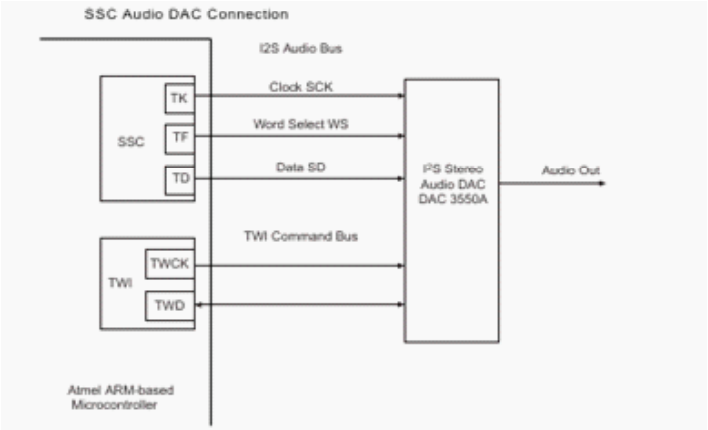
برای اطلاعت بیشتر به ادرس زیر مراجعه کنید :

http://en.wikipedia.org/wiki/CAN_bus



((Synchronous Serial Controller (SSC

کنترل کننده سریال هم زمان ، تطابق داده ارسالی و دریافتی را در انواع پروتکل های سریال کنترل میکند (i۲c ، i۲s و...) و در صورت عدم تطابق داده دریافت شده توسط گیرنده و داده ارسال شده توسط فرستنده ، میکرو و کاربر را خبر دار میکند .



مثال :DAC ۳۵۵۰A یک مبدل داده i۲c به داده انالوگ میباشد ، شما میتوانید توسط باس i۲c انواع صدا های دیجتال ، ولتاژ های دیجتال و.... را به این dac بدهید و در خروجی ان سیگنال انالوگ را در یافت کنید .

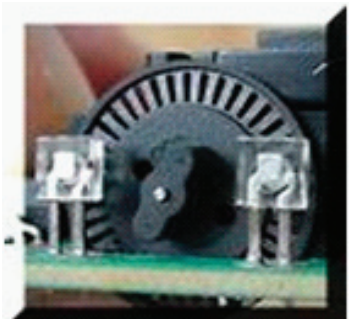
در این dac سرعت تبادل داده بسیار زیاد میباشد و باس i۲c نمیاتواند سیگنال خطا را رد یابی کند ، به همین دلیل در این ایسی باس ssc تعبیه شده است :

اطلاعات دیجتال از طریق i۲c ارسال میشود و صحت ان توسط ssc کنترل خواهد شد ، این عملیات سرعت بالایی را در بخش صدا حاصل میکند (قابلیت بخش صدای ۳۲ بیتی (با فرکانس ۸ تا ۴۸۰۰۰ هرتز) .

پشتیبانی از ISO۷۸۱۶ T۰/T۱ Smart Card

ISO۷۸۱۶ یک پروتکل استاندارد برای تبادل اطلاعات بین چیپ های الکترونیکی و پردازنده ها میباشد .چیپ های الکترونیکی ، بر روی کارت های از جنس پلاستیک نصب میشود و معمولا شما انها را به نام اسمارت کارت میشناسید ، کارت های تلفن و کارت های اعتباری نمونه های مختلف از اسمارت کارت هستند .

پردازنده نیز وظیفه دریافت اطلاعات از کارت و تبدیل ان به داده قابل فهم برای انسان یا ارسال ان به پردازنده مرکزی را به عهده دارد .

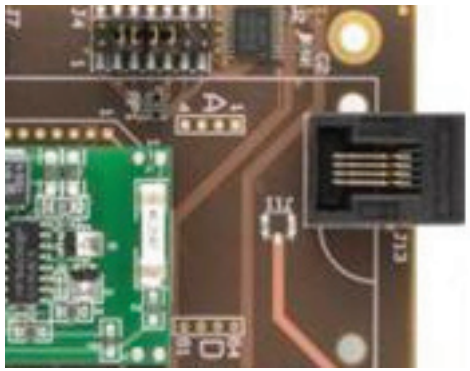


پشتیبانی از IrDA ® Infrared Modulation/Demodulation

تاکنون led های مادون قرمز را مشاهده کرده اید :

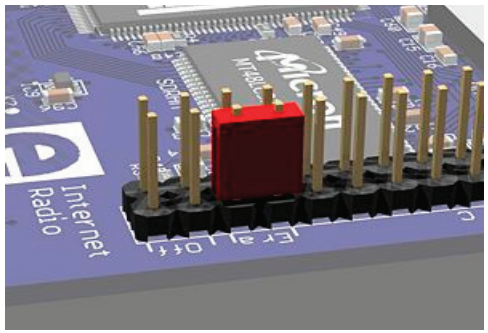
شاید فکر کنید از این led ها فقط برای تشخیص مانع و ... استفاده میشود اما چنین نیست .

در الکترونیک میتوانیم از این نوع led ها برای ارسال و دریافت داده استفاده نماییم ، میکرو کنترلر های arm به شما این امکان را میدهند تا داده خود را به فرم دلخواه مدوله کنید و سپس ان را با این نوع led ها در فضا انتشار دهید . در میکرو گیرنده شما بعد از دریافت این امواج باید ان را دمودله کنید و داده خود را تحویل بگیرد .



پشتیبانی از خط مودم . (Modem Line)

مودم برای انتقال اطلاعات بین کامپیوترها از طریق کانال های مخابراتی به کار می رود. یک مودم در سر راه خطوط ارسالی قرار می گیرد تا بتواند پالس های دیجیتال را به سیگنال های آنالوگ تبدیل کند. سپس سیگنال های آنالوگ را می توان از طریق خطوط تلفن . فیبر نوری ، کابل کواکسیال، مایکروویو ، ماهواره مخابراتی و غیره انتقال داد.. کامپیوتری که می بایست اطلاعات را دریافت کند نیز از یک مودم دیگر استفاده می کند تا بدین وسیله سیگنال آنالوگ را مجدداً به صورت پالس های دیجیتال ، تبدیل کند که این عمل را دمودله گویند. شرکت اتمل این امکان را برای شما مهیا کرده است که به جای کامپیوتر از میکرو کنترلر arm استفاده کنید ، این میکرو اطلاعات موازی خروجی مودم را میخواند و میتواند تصمیمات لازم را اتخاذ نماید ، این قابلیت دو طرفه میباشد (میکرو قابلیت ارسال و دریافت اطلاعات را دارد) .



پروگرام کردن تعداد زیادی از انها بدون نیاز به سخت افزار پروگرامر و از طریق پورت usb

به نظر من یکی از بهترین امکانات میکرو کنترلر های arm وجود Boot ® SAM-BA در انها میباشد .

توسط Boot ® SAM-BA میتوانید میکرو کنترلر را از طریق پورت usb و بدون نیاز به سخت افزار پرگرامر به راحتی پرگرام کنید. در ادمه به تشریح کامل این مورد پرداخته ایم .

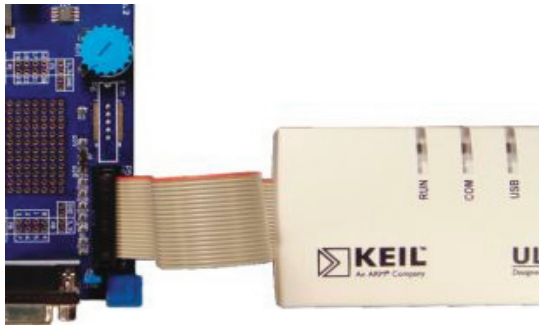
تمامی میکرو کنترلر های خانواده arm از واسط jtag پشتیبانی میکنند .

IEEE Standard ۱۱٤۹.۱–۱۹۹۰ Test Access Portand Boundary–Scan Architecture یا jtag یک پروتکل ارتباطی میباشد که توسط تعدادی از شرکت های وابسته به سازمان ieee و تحت استاندارد ان به ثبت رسیده است .در پروتکل jtag دسترسی کامل به cpu و حافظه ها فراهم میباشد ، و شما میتوانید داده های پردازش شده یا در حال پردازش توسط انها را مشاهده کنید .

رابط jtag از ٤ پایه اصلی برای ارتباط با سخت افزار استفاده میکند :

هر وسیله ای که با استاندارد Jtag سازگار باشد لازم است تا پین های زیر را داشته باشد :

۱– Clock : (Test Clock Input) TCK ورودی ، این پالس برای همزمانی میان دستگاه مورد تست و پروگرامر jtag میباشد .



۲– TDI (Test Data In) : از طریق این پایه ، داده از پروگرامر به دستگاه در حال تست وارد میشود .

۳– TDO (Test Data Out): از طریق این پایه ، داده از دستگاه مورد تست به پروگرامر میرود .

٤– TMS (Test Mode Select) : از طریق این پورت حالتهای مختلف تست انتخاب می شود.

همچنین در این میان دو پایه دیگر نیز وجود دارد :

۵– TRST (Test Reset Input) این پایه از پروگرامر به ریست دستگاه متصل میشود و قبل از شروع کار ان را باز نشانی میکند .

٦– JTAGSEL (JTAG SELECT) : برای راه اندازی پروتکل JTAG ، این پایه باید یک شود .

برای کار با JTAG به موارد زیر نیاز دارید :

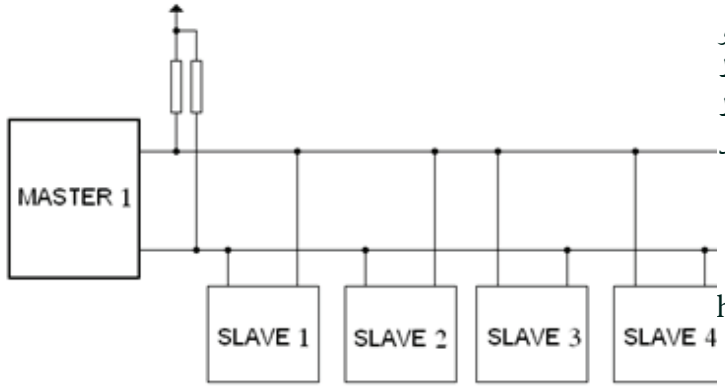
کابل ارتباطی – سخت افزار JTAG – نرم افزار JTAG

کابل ارتباطی وظیفه اتصال دستگاه مورد تست را به پروگرامر را به عهده دارد ، معمولا پروگرامر از طریق یک کابل دیگر به پورت سریال کامپیوتر متصل میشود .

سخت افزار JTAG وظیفه کنترل داده های ارسالی و دریافتی و همچنین تغییر دادن انها به کد قابل فهم کامپیوتر را به عهده دارد .وظیفه نرم افزار JTAG ، تبدیل کد های دریافتی به علائم نمایشی و کد های قابل فهم برای انسان میباشد .

با توجه به تنوع سخت افزار و نرم افزار پروگامر های JTAG زیادی برای کار با میکرو کنتر لرهای ARM ارئه شده است ، شما میتوانید نقشه ها و اطلاعات بیشتر را در سایت WWW.ATMEL.COM بیاید .

پشتیبانی از پروتکل معروف Two-wire (i۲c)

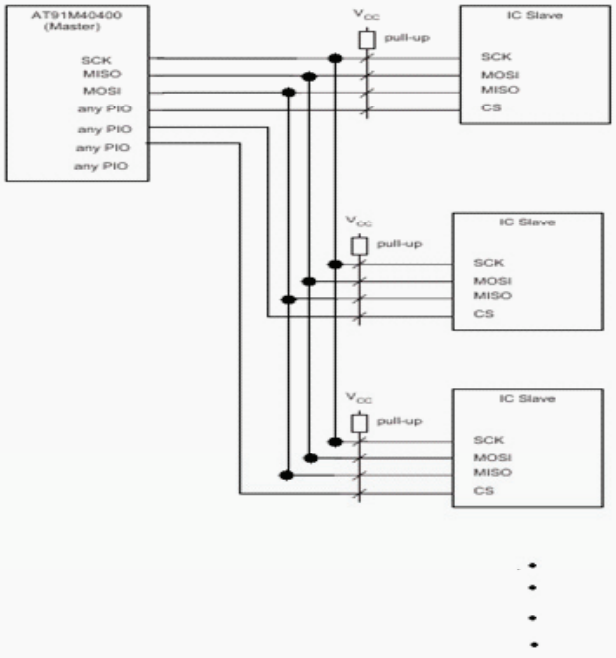


پروتکل Two-wire یا ارتباط دو سیمه ، یک پروتکل ارتباطی سریال میباشد . در این پروتکل از دو سیم برای تبادل داده و کلاک ، و سیم های گراند و vcc استفاده میشود . در این پروتکل میتوان توسط دو سیم تعداد ۱۲۸ وسیله را به هم متصل نمود ، هر وسیله دارای یک ادرس منحصر بفرد میباشد و توسط همان ادرس شناسایی میشود . در این پروتکل میگرو فرستنده به عنوام مستر و میکرو گیرنده اسلیو میباشد . برای دریافت اطلاعات بیشتر در مورد i۲c به ادرس زیر مراجعه کنید :

http://forum.ir-man.com/showthread.php?tid=١٦٤

پشتیبانی از پروتکل spi

Serial Peripheral Interface یا پروتکل spi یک رابط سریال برای انتقال داده به صورت همزمان بین دو وسیله میباشد ،اکثر میکرو کنترلر های arm د دارای دو عدد خط ارتباطی SPI مجزا میباشد ، SPIA از خطوط SPCKA (خروجی کلاک از مستر به اسلیو) و MISOA (داده خروجی از اسلو به مستر (دیتا به این خط وارد میشود)) و MOSIA (داده خروجی از مستر به اسلیو (داده از این خط خارج میشود) و NSSA (انتخاب کننده اسلیو ۱) استفاده میکند ، برای SPIB نیز از خطوط SPCKB و MISOB و MOSIB و NSSB استفاده میشود .



هنگامی که در برنامه پروتکل SPI راه اندازی میشود ، میکرو Master شروع به ارسال پالس به میکرو Slave میکند ، ارسال پالس از طریق خط SPCK انجام میشود ، این پالس برای همزمانی میان دستگاه ها میباشد .با یک شدن پایه NSSA میکرو Slave انتخاب میشود و دو میکرو شروع به تبادل داده با هم میکنند .پروتکل SPI این قابلیت دارد که در هر واحد زمانی با یک میکرو به تبادل اطلاعات بپردازد ،برای فعال سازی دستگاه های بیشتر از خطوط NPCSA۰ NPCSB۰ – NPCSB۳ – NPCSA۳ استفاده میشود.

همانطور که در تصویر مشاهده میکنید ، مدار ما دارای یک میکرو مستر و تعداد سه عدد اسلیو میباشد ، با فعال شدن اولین اسلو میتوانیم داده را به ان منتقل کنیم یا از ان بگیریم و...

فعال شدن اسلو های از طریق پایه chip select (cs) انجام میشود ، شما میتوانید از هر پایه های به عنوان cs استفاده کنید . معمولا از پایه های NPCSA۳ – NPCSA۰ NPCSB۳ – NPCSB۰– که برای این منظور ایجاد شده انداستفاده میشود

پشتیبانی از پروتکل rs۲۳۲

اغلب میکرو کنترلر های arm دارای دو پورت مجزا برای rs۲۳۲ میباشند .

در این ارتباط از دوسیم به نام های rxd و txd استفاده میشود که خط rxd وسیله اول دیتا را از ان به بیرون منتقل میکندو خط خروجی دیتا است و به ورودی دیتا دستگاه دوم (txd) متصل میشود و خط txd ورودی دیتا است که به خروجی دیتای دستگاه دیگر (rxd) متصل میشود.

این پروتکل تقریبا در تمامی وسایل الکترونیکی استاندار استفاده میشود ، ما در کامپیوتر ان را به نام پورت com میشناسیم .

وجود واحد Debug Unit)

توسط این واحد شما میتوانید عملیات زیر را انجام دهید :

•application software

•operating systems

•hardware systems based on an ARM processor.

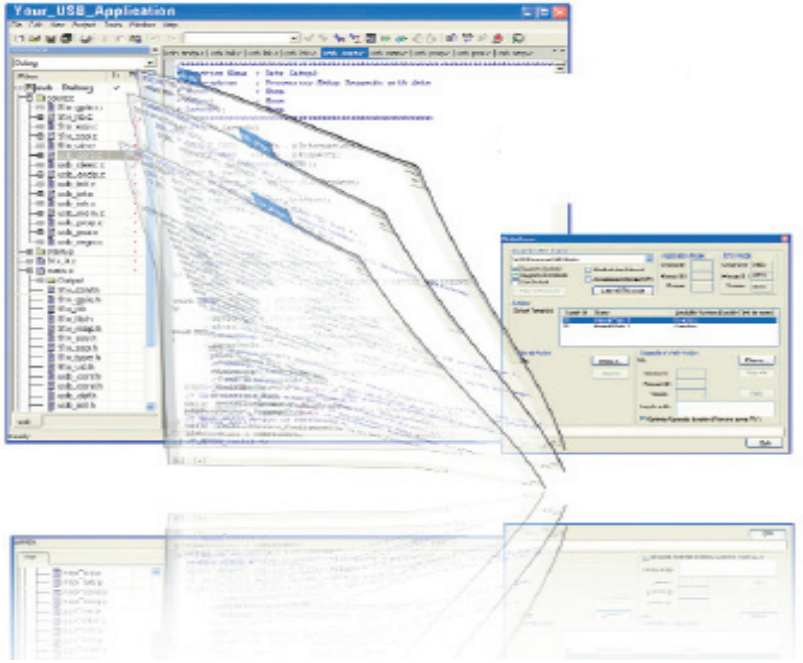
•The debug unit enables you to:

•stop program execution

•examine and alter processor and coprocessor state

•examine and alter memory and input/output peripheral state

•restart the processor core.

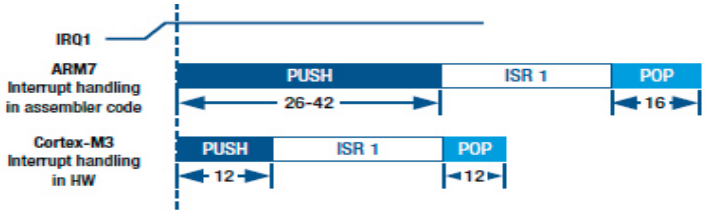


دارای چندین منبع وقفه :

وقفه یا Interrupt یکی از امکانات کاربردی میکرو کنترلر های ARMمیباشد ، با استفاده از وقفه م

برنامه وقفه میرود و برنامه دیگری را را اجرا میکند .

میکرو کنترلر های arm دارای دو منبع وقفه خارجی میباشند :



- FIQ – Fast Interrupt
- IRQ – Normal Interrupt

نوع اول وقفه سریع میباشد ، هنگامی که یک پالس به پایه مربوط به این وقفه اعمال میشود (سطح بالس در برنامه مشخص میشود)، cpu میکرو

تمام کار های خود را رها میکند و زیر برنامه وقفه میرود .

نوع دوم وقفه معمولی میباشد ، در این نوع وقفه cpu میکرو دستور در حال اجرا را تمام میکند و به زیر برنامه وقفه میرود .cpu بعد از انجام

دادن زیر برنامه وقفه ، به حلقه اصلی برمیگردد و برنامه قبلی را ادامه میدهد .

در میکروکنترلر های Arm منابع وقفه داخلی نیز وجود دارد که در اموزش برنامه نویسی به بررسی انها پرداخته میشود .

یکی دیگر از امکانات ویژه میکرو کنترلر های arm پشتیبانی از واسط i۲s میباشد .

Inter–IC Sound یا رابط i۲s یک پرتکل سه سیمه میباشد که امکان ورود صدای دیجیتال به میکرو کنترلر را محیا میکند همچنین میکرو میتواند صدای دیجیتال را از طریق این پروتکل به dac های مخصوص ارسال کند .

The three lines defined by the I۲S pro tocol are:

- a Serial Data (SD) line c ontaining two time–division multiplexed channels
- a left/right channel Word Select (WS)
- a continuous Serial Clock (SCK)

یک نمونه از ایسی های که برای تبدیل صدا انالوگ به دیجیتال و برعکس کار برد دارد چیپ TSC۲۳۰۱ میباشد ، همچنین ایسی dac۳۵۵۰A صدای دیجیتال را به انالوگ تبدیل میکند . در اموزش برنامه نویسی با i۲s بیشتر اشنا خواهیم شد .

معمولا دستگاه های همچون cdrom و dvdrom و ... دارای خروجی صدای دیجتال میباشند .

میکرو کنترلر های ها arm مجهز به واحد های MC و RSTC و PMC و AIC و... میباشند ، وجود این واحد باعث مصونیت میکرو در برابر نویز میشود .

این واحد نقش کنترل کردن واحد های مربوط به خود را به عهده دارند . مثلا واحد MC یا Memory Controller وظیفه کنترل کرد حافظه فلش و رفع خطا های احتمالی ان را به عهده دارد .

واحد RSTC یا Reset Controller : این واحد وظیفه باز نشانی میکرو در شرایط اضطراری همچون کم شدن ولتاژ تغذیه و... را به عهده دارد . این واحد امکان باز نشانی میکرو از طریق پایه nrst (ریست دستی) را نیز فراهم اورده است .

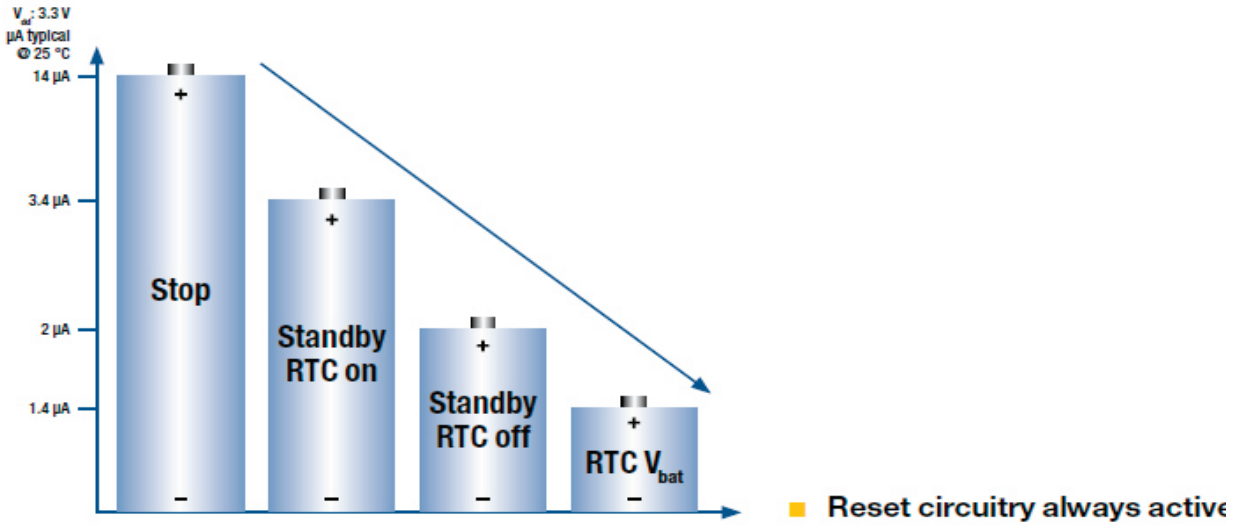
واحد PMC یا Power Management Controller : با فعال بودن این واحد ، میکرو میتواند عملکرد خود را با توجه به ولتاژ تغذیه تنظیم کند . میکرو میتواند در شرایط کمبود ولتاژ و جریان تغذیه ، فرکانس کاری خود را تا زیر ۵۰۰ هرتز کاهش دهد .

واحد AIC یا Advanced Interrupt Controller : یکی از موارد مهم در قطعات پرسرعت ، وقفه های میباشند . همانطور که قبلا گفتیم ، در شرایط وقفه میکرو برنامه در حال اجرا را رها کرده و به زیر برنامه وقفه میرود (در برنامه نویسی بیشتر با این مورد اشنا خواهیم شد .) برای جلوگیری از هر گونه تاخیر در اجرای زیر برنامه ، این واحد به کمک شما می اید .

میکرو کنترلر هایarm دارای تعداد زیادی خطوط I/O میباشند ، این خطوط در تمامی میکرو کنترلر ها قابل برنامه ریزی اند . برای مثال ، میکرو AT۹۱SAMvX۲۵٦ دارای دو پورت میباشد ، این دو پورت تعداد ٦۲ خط ورودی و خروجی قابل برنامه ریزی را در اختیار شما می گذارد .

میکرو کنترلر های ARM دارای تایمر کانتر ها ١٦ تا ٢٤ بیتی میباشند . همچنین در این میکرو ها خروجی های ١٦ PWM بیتی برای کار کنترلی و کار با صوت ایجاد شده است .

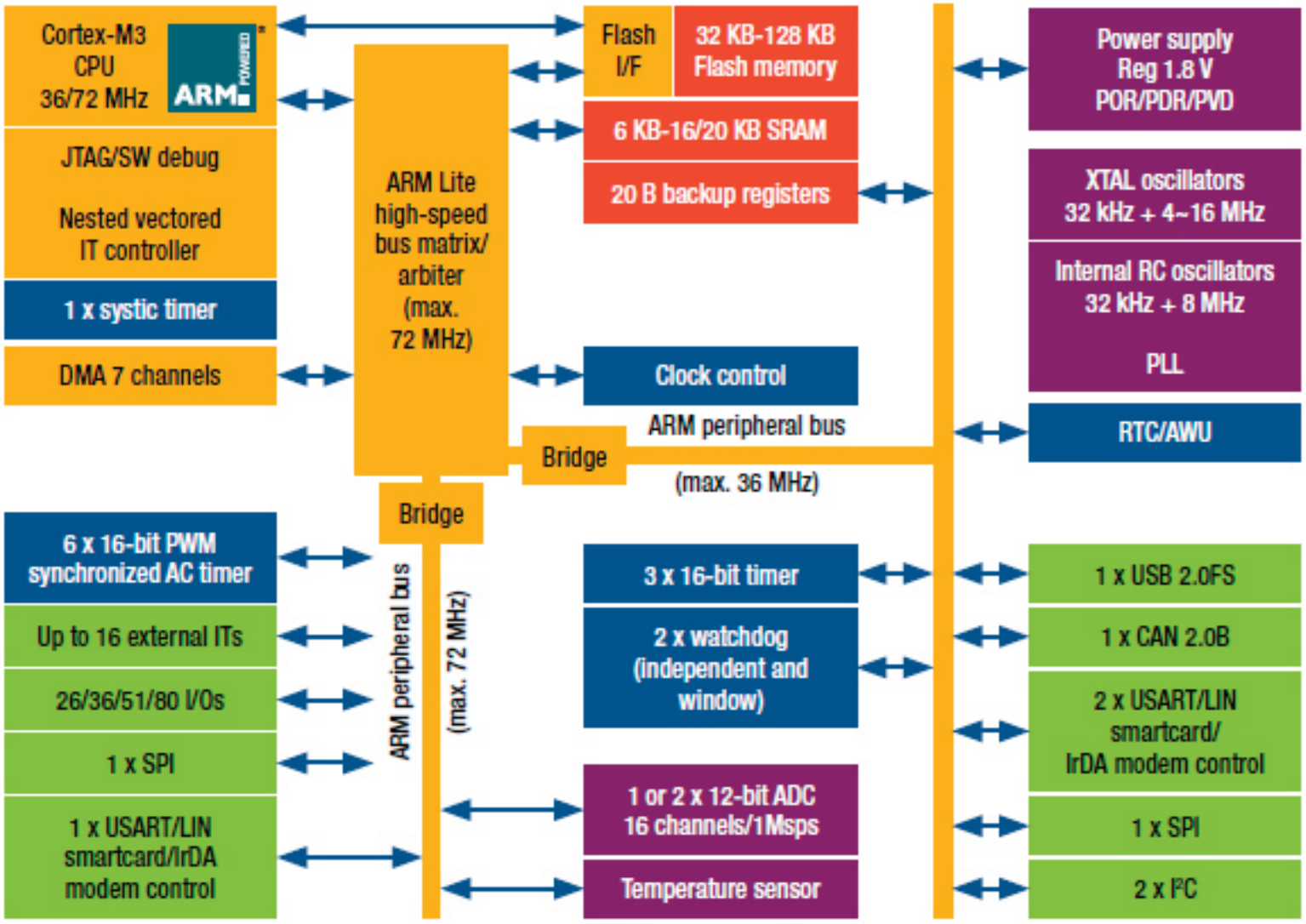
میکرو کنترلر های arm بسیار کم مصرف میباشند . همانطور که قبلا گفتیم واحد PMC یا Power Management Controller وظیفه تنظیم کردن مصرف توان میکرو را به عهده دارد . مد های کم مصرف یا sleep میکرو کنترلر های arm گاهی به ۷ مورد میرسد . در بعضی از میکرو کنترلر ها جریان مصرفی میتواند تا زیر ۱ میکرو امپر نیز کاهش یابد



از دیگر امکانات این میکرو کنترلر ها میتوان به وجود مبدل های انالوگ به دیجتال ۱۰ تا ۱۶ بیتی ، RTC داخلی ، .... اشاره کرد .شما میتوانید سایر اطلاعات را در دیتا شیت میکرو کنترلر ها بیابید .

در صفحه بعد بلوک دیاگرام یکی از میکرو کنترلر های arm اورده شده است . همانطور که مشاهده میکنید هسته اصلی که توسط گروه arm طراحی شده Cortex-M۳ میباشد . شرکت سازنده میکرو کنترلر به این هسته امکاناتی از قبیل پورت های usb و can و uart و... اضافه میکند و ان را به عنوان یک میکرو کنترلر جدید به فروش میرساند .

در دیتا شیت این میکرو کنترلر ها بلوک دیاگرام و نام هسته میکرو وجود دارد . بازنگری بلوک ها شما را در درک مطالب یاری میکند .



DMA: Direct memory access
RTC: Real-time clock
AWU: Auto wake-up capability with RTC alarm

POR: Power-on reset
PDR: Power-down reset
PVD: Programmable voltage detector

کامپایلر ها و مفسر های ARM

برای ARM کامپایلر ها و مفسر های زیادی ارائه شده است ، زبان بر نامه نویسی اغلب این کامپایلر ها C و C++ میباشد ، در زیر نام این کامپایلر ها را اورده ایم :

: IAR for ARM

قابلیت برنامه نویسی میکرو کنترلر های ARM به زبان های C و C++ و اسمبلی / امکان شبیه سازی برنامه نوشته شده / پشتبانی از تمامی میکرو کنترلر های ARM / دارای منابع اموزشی متوسط./محیط حرفه ای و استفاده از ویرایش گر قوی

: CrossWorks for ARM

قابلیت برنامه نویسی میکرو کنترلر های ARM به زبان اسمبلی و C / پشتیبانی از میکرو کنترلر های که از ARMv استفاده میکنند / محیط و ادیتور ساده / عدم شبیه سازی برنامه نوشته شده / دارای منابع اموزشی کم .

: keil arm

قابلیت برنامه نویسی میکرو کنترلر های ARM به زبان های C و C++ و اسمبلی / امکان شبیه سازی برنامه نوشته شده / پشتبانی از تمامی میکرو کنترلر های ARM / دارای منابع اموزشی متوسط./محیط حرفه ای و استفاده از ویرایش گر قوی /یادگیری سریع نرم افزار /

: WinArm

قابلیت برنامه نویسی میکرو کنترلر های ARM به زبان های C و C++ / عدم شبیه سازی برنامه نوشته شده / پشتیبانی از میکرو کنترلر های که از ARMv استفاده میکنند / دارای منابع اموزشی متوسط./محیط حرفه ای و استفاده از ویرایش گر قوی /متن باز بودن نرم افزار و سایر امکانات جانبی

: Flowcode ARM

قابلیت برنامه نویسی میکرو کنترلر های ARM به زبان گرافیکی (بلوگ دیاگرامی) / پشتیبانی از میکرو کنترلر های که از ARMv استفاده میکنند / دارای منابع اموزشی متوسط./ محیط ساده /امکان شبیه سازی برنامه نوشته شده / یادگیری اسان

: ARM ADS

قابلیت برنامه نویسی میکرو کنترلر های ARM به زبان های C و C++ و اسمبلی / امکان شبیه سازی برنامه نوشته شده / پشتبانی از تمامی میکرو کنترلر های ARM / دارای منابع اموزشی متوسط./محیط حرفه ای و استفاده از ویرایش گر قوی/

سایر کامپایلر ها :

www.micrium.com

www.smxrtos.com

www.segger.com

ما ایرانی ها از نرم افزار های مختلف ، انتظارات زیر را داریم :

۱- بتوانیم اخرین ورژن ان را به صورت رایگان دانلود کنیم (اخرین ورژن با کرک یا سریال معتبر)

۲- بسرعت ان را یادبگیریم و بتوانیم با ان کار کنیم .

۳- از تمامی میکرو کنترلر ها و امکانات جانبی انها پشتیبانی کند .

٤- دارای سورس ، پروژه ، منابع اموزش و.... زیادی باشد .

۵- حجم کد هگز خروجی کم باشد (بهینه باشد) و در برنامه نویسی باگ یا خطایی نداشته باشد .

٦- با یادگیری ان ، نیاز ما کاملا رفع شود و برای کار با میکرو کنترلر های جدید (که در اینده عرضه میشوند) با مشکلی روبرو نشویم .

۷- و....

جواب سوالات :

سوال ۱ : تقریبا تمامی کامپایلر بالا به جز ads , iar (معمولا کرک این دو کامپایلر برای ورژن های جدید موجود نیست)

سوال ۲ : تقریبا تمامی موارد

سوال ۳ : keil و iar و ads

سوال ٤ :winarm

سوال ۵ :نرم افزار keil داری بهینه ترین خروجی میباشد . در این بین win arm رتبه دوم و ads رتبه سوم را داراست .

سوال ٦ : keil و iar و ads

نکته ها :

۱- محیط کامپایلر Flowcode for ARMs کاملا گرافیکی میباشد و شما میتوانید در کمترین زمان ان را یاد بگیرید ، این کامپایلر از میکروکنترلرهای که ازهسته arm۷ استفاده میکنند پشتیبانی میکند (تعداد محدودی از میکرو کنترلر ها atmel) و همانطور که میدانید در زبان های متنی قدرت مانور بیشتری در برنامه نویسی وجود دارد (در گرافیک شما به المان های موجود محدود میشوید)

۲-کامپایلر winarm که یکی از کامپایلر های رایگان برای arm میباشد، نزدیک به سه سال است که ابدیت نشده، به نوعی میتوان گفت که این کامپایلر به حال خود رها شده و اینده ای ندارد .

۳-کامپایلر CrossWorks for ARM ظاهرا فقط از ۷ arm پشتیبانی میکند



٤- IAR for ARM و arm ads از قدرتمندترین کامپایلر های arm میباشند ، اما نسخه کامل یا کرک معتبر این نرم افزار ها به سادگی در دسترس ما قرار نمی گیرد ، ممکن است نسخه مناسب (برای ما ایرانی ها) چند ماه بعد از عرضه نسخه اصلی ارائه شود ، این نسخه کرک شده نسخه دمو است و به درستی کار نمیکند .

۵-همانطور که مطرح شد ، برای arm کامپایلر های دیگری نیز وجود دارد . در این اموزش ما به بررسی کامپایلر های معروف پرداختیم . در اغلب انجمن ها و سایت ها در مورد این کامپایلر ها بحث شده است و شما میتوانید در اینترنت انها را جستجو کنید .

با توجه به مطالب گفته شده در بالا بهترین کامپایلر برای ما keil arm هست. معمولا اخرین ورژن این نرم افزار به همراه کرک و سریال معتبر، همیشه در دسترس ما قرار دارد ، و دوستانی که با ان کار میکند تا کنون مشکلی را گزارش نکرده اند .

میکرو کنترلر های موجود در ایران

قبل از بررسی محیط این کامپایلر و اموزش نحوه برنامه نویسی در ان ، به مقایسه ای اجمالی بین میکرو کنترلر های مختلف پرداخته ایم :

در ایران خانواده های زیر موجود می باشد :

atmel : سری ۷x و ۹x (این میکرو کنترلر های به ترتیب از هسته arm۷ و arm۹ ستفاده میکنند)مانند: AT۹۱SAM۷X۱۲۸وAT۹۱SAM۷X۲۵۶ و AT۹۱SAM۹۲۶۰ و AT۹۱RM۹۲۰۰ جدیدا هم مشاهده شده برخی وارد کنندگان سری قدیمی ۷s را وارد می کنند و این سری مثال Tiny در مقابل Mega در خانواده AVR هست

phillips : میکرو های ساخته شده توسط شرکت فیلیپس نیز در ایران وجود دارد :LPC۲۱۴۶ و LPC۲۳۶۸ و LPC۲۳۷۸

STMicroelectronics : شرکت ST نیز اقدام به تولید میکرو کنترلر های با هسته ARM نموده است ، در ایران میکرو کنترلر های STM۳۲F۱۰۱T۶ و STM۳۲F۱۰۱CB و... موجود است .

شرکت ها دیگری نیز اقدام به تولید میکرو کنترلر ازهسته ARM نموده اند که بحث ما برسر میکرو کنترلر های موجود در ایران میباشد .

بحث بر سر برتری میکرو کنترلر ها نیز هیچگاه به پایان نمیرسد . ما برای کار با ARM میکرو کنترلر های ساخته شده توسط اتمل را به دلایل زیر انتخاب کرده ایم :

این شرکت با عرضه میکرو کنترلر های AVR و ۸۰۵۱ سابقه درخشانی در ایران دارد .

پشتیبانی از تولیدات ، در این شرکت بی نظیر است .

محصولات این شرکت بیشتر در ایران وجود دارد (به نوعی همه گیر شده است)

از دیگر ویژگی ها میتوان به تنوع محصولات ، قیمت کم ، وجود منابع اموزش در سایت رسمی شرکت و فروم های خارجی ، وارادات اسان و تامین قطعات جانبی و.... اشاره کرد .

مطالبی که در اینترنت می بینیم :

محصولات شرکت phillips از سرعت بیشتری نسبت به میکرو کنترلر های ARM برخوردارند :

این دیدگاه را میتوان ناشی از تعصب به میکرو دانست ، این فرد فقط با این خانواده کار کرده است و به نوعی معتقد است دیگران نیز باید به سمت این میکرو بیایند . در حالی که حداکثر سرعت میکرو در دیتاشیت ان درج شده است و میکرو های مختلف دارای سرعت ها مختلفی میباشند .

از میکرو کنترلر های سامسونگ در گوشی های همراه استفاده شده است . (هر میکرو کنترلری که در یک مکان مهم استفاده شود بهتر است)

معمولا شرکت های سازنده گوشی ، از میکرو کنترلر های خودشان استفاده میکنند ، شرکت سامسونگ یکی از تولید کننده گان مطرح گوشی های همراه میباشد و استفاده از این میکرو در گوشی های این شرکت طبیعی است و دلیل بر قدرت بیشتر این میکرو نیست .

در این میکرو ها از فناوری AVR استفاده شده و این میکرو در برابر نویز ضعیف میباشد .

در شرکت اتمل واحد های طراحی و ساخت هر محصول کاملا مجزا میباشد ، به علاوه مهمترین بخش که باید در برابر نویز ایمن باشد CPU است که

توسط ARM طراحی شده است ، به علاوه وجود واحد های کنترلی که قبلا به انها اشاره شد ، تاثیر نویز را تا حد زیادی کاهش میدهد .

نکته بسیار مهم

۱–از انجا که در نرم افزار های برنامه نویسی ، تبدیل کد های C به کد هگز بر اساس فایل های هدر (STARTUP) انجام میشود و این فایل ها توسط کمپانی سازنده میکرو ارائه شده است . هر خانواده دارای ریجستر های مخصوص به خود میباشد . مثلا اگر شما برنامه نویسی را با میکروکنترلر های اتمل شروع کنید ، در اینده نمیتوانید از دانش خود برای میکرو کنترلر های phillips استفاده کنید . پس از همین جا ، بهترین میکرو را انتخاب کنید و ان را یاد بگیرید .پس در اولین قدم همراه ما باشید

۲–در زبان c برای کار با هر پروتکل یا وسیله جانبی به یک کتابخانه نیاز است ، کتابخانه یا هدر فایل کد های اصلی که برای راه اندازی پروتکل مورد نظر نیاز است را به زبانی شیوا تر تبدیل میکند ، در واقع با وجود فایل های هدر شما کمتر با سخت افزار درگیر میشوید و نیازی به درک ان ندارید .

در نرم افزار keil تعدادی از فایل های هدر وجود ندارد (به دلیل دمو بودن نرم افزار) می میتوانیم این فایل ها را از اینترنت گیر بیارویم یا خودمان انها را بنویسیم .

سایر نرم افزار های برنامه نویسی نیز فاقد فایل های هدر میباشند و این مورد فقط مربوط به keil نیست .

برای در ک بهتر موارد بالا ، مثال های صفحه بعد را ببینید :

```
#include <AT91SAM7X256.H>                /* AT91SAMT7X256 definitions */
#include <lib_AT91SAM7X256.h>
void wait (void) {
 unsigned int n;
 for (n = 0; n < 7372800; n++);
}
int main (void) {
 AT91F_PMC_EnablePeriphClock(AT91C_BASE_PMC, 1 << AT91C_ID_PIOB);
 AT91F_PIO_CfgOutput(AT91C_BASE_PIOB, AT91C_PIO_PB19);
 for (;;) {
    AT91F_PIO_ClearOutput(AT91C_BASE_PIOB, AT91C_PIO_PB19);
    wait();
     AT91F_PIO_SetOutput  (AT91C_BASE_PIOB, AT91C_PIO_PB19);
     wait();
}
}
```

```
#include <LPC21xx.H>
void delay(void);
int i;
void main() {
IODIR0 = 0xFF;
while (1({
IOSET0 =0x01;
Delay();
IOCLR0 = 0x01;
Delay();
 }
}
void delay()
{
```

هر دو برنامه یکی از پایه های میکرو خود را خاموش روشن میکنند ، اما انچه در این برنامه متفاوت مباشد ، ریجستر های مربوط به پیکربندی پورت ها است ، این ریجستر برای تمامی امکانات با هم متفاوت خواهند بود . برنامه سمت راست برای میکرو کنترلر AT۹۱SAM۷X۲۵٦ اتمل و برنامه سمت چپ برای میکروی LPC۲۱٤٦ فیلیپس نوشته شده است (فعلا با کد ها کاری نداشته باشید ف در ادامه با تمامی انها اشنا می شویم)

برای شروع کار میکرو کنترلر AT۹۱SAMvX۲۵٦ از خانواده اتمل را انتخاب نموده ایم ، ما برای این میکرو یک بردآموزشی ساخته ایم در ادامه مشخصات و نحوه تهیه برد اورده شده است .
دلیل انتخاب این میکرو ، وجود ان در اکثر فروشگاه های ایران و قیمت مناسب (با توجه به امکانات) بود در بخش ضمائم ، دیتا شیت فارسی این میکرو اورده است .
اموزش ها و سورس کد های ارائه شده از طرف ما برای این میکرو میباشد ، شما میتوانید به سادگی سورس ها برای کار با دیگر میکرو کنترلر های اتمل تغییر دهید (کافی است نام میکرو و پایه ها و... را عوض کنید)

در راه اموزش این میکرو کنترلر به زبان فارسی ، اقدام به تولید یک برد اموزشی نمودیم . برد اموزشی تولید شده دارای دو بخش میباشد :

۱-برد میکرو (Cpu Board)

۲- برد اصلی

برد میکرو :

در این برد ، پایه های میکرو کنترلر at۹۱sam۷x۲۵٦ را به دو سوکت IDC ٤۰ پایه منتقل کرده ایم ، همچنین بر روی برد امکاناتی برای برنامه ریزی و تست برد وجود دارد .

برد اصلی :

در این برد امکاناتی نظیر lcd رنگی ، پورت های com ، سوکت مخصوص مموری ، کلید های فشاری ، led و.... تعبیه شده است . این برد توسط دو عدد سوکت IDC ٤۰ پایه با برد میکرو ارتباط بر قرار میکند .(شرح کامل امکانات در ادامه اورده شده است)

شما میتوانید به کابل های IDC سوکت های نری متصل کنید و پایه های مورد نیاز را بر روی برد برد بیاورید یا برد اصلی را تهیه کنید و کار خود را راحت کنید .

عرضه برد به روش بالا دارای مزیت های زیر است :

۱- در صورتی که میکرو اسیب ببنید ، شما فقط یک برد میکرو از دست می دهید (نیازی به لحیم کاری مجدد و.... نیست ، برد معیوب را به راحتی تعویض میکنید)

۲- به راحتی میتوانید دیگر میکرو کنترلر های arm را نیز با برد اصلی راه اندازی کنید (در اینده برد میکرو ، مخصوص دیگر میکرو کنترلر های ARM و PIC و AVR طرحی و عرضه میشود)

۳- شما میتوانید با توجه به توان خود هزینه کنید ، در صورتی که قبلا با میکرو کنترلر ها و امکانات انها کار کرده اید ، برد میکرو مناسب کار شماست و نیازی به پرداخت هزینه ی اضافه نیست ، در صورتی که قبلا با میکرو کنترلر ها کار نکرده اید ، میتوانید هر دو برد را تهیه کنید ، با این کار امکان اسیب رسیدن به میکرو و دیگر قطعات به شدت کم میشود .

٤- شما میتوانید به سلیقه خود برد دیگر را تولید کنید ، برد میکرو را تهیه کنید ، بعد از اینکه مقداری با arm کار کردید ، برد دوم را با توجه به امکانات مورد نیاز خود تهیه کنید .

۵- شما میتوانید از برد اصلی برای راه اندازی دیگر میکرو کنترلر ها (خانواده های AVR۳۲ ، PIC و... که در اینده به اموزش انها نیز می پردازیم) استفاده کنید .

امکانات برد :

۱- LCD رنگی

۲- LCD کارکتری

۳- سوکت مخصوص حافظه های MMC/SD

٤-دارای تعداد ٤ عدد کلید فشاری (متصل به پایه های وقفه)

۵ - دارای چهار عدد LED (متصل شده خروجی های PWM)

٦- ورودی مجزا برای اتصال میکرو فن (شما میتوانید با ولتاژ های انالوگ و به راحتی کار کنید)

۷- خروجی مجزا برای صدا (متصل به بلند گو)

۸- بازر

۹-مجهز به حافظه EEPROM سریال برای کار با پروتکل I۲C

۱۰- مجهز به چیپ DS۱۳۰۷ (RTC)

۱۱- درای دو پورت COM .

۱۲ - خروجی تبدیل شده واحد DEBUG UNIT (قابلیت اتصال مستقیم به کامپیوتر)

۱۳- خروجی CAN

۱٤ - مبدل ولتاژ مجزا برای بک لایت LCD رنگی

۱۵- دارای ایسی تقویت کننده جریان (ULN۲۰۰۳) برای راه اندازی رله و موتور

۱٦ - دارای پورت JTAG مجزا

۱۷- دارای پورت USB مجزا و قابلیت پروگرام کردن میکرو از طریق رابط SAM-BA

۱۸- مجهز به سنسور LM۳۵ برای کار با واحد ADC

۱۹- نصب یک پتانسیومتر برای اعمال ولتاژ انالوگ به ADC

۲۰- خروجی مجزا برای ADC۰ تا ADC۷

۲۱- خروجی مجزا برای SPI۰ و SPI۱

۲۲- خروجی مجزا برای UART۰ و UART۱

۲۳- خروجی مجزا برای خروجی های PWM و پایه های وقفه

۲٤-خروجی مجزا برای کار با پروت I۲S

۲۵- داری پورت کیبرد و موس کامپیوتر

۲٦- تغذیه مجزا برای برد اصلی و میکرو (شما میتوایند از برد به صورت مجزا استفاده کنید)

۲۷-طراحی برد اصلی در یک لایه (امکان اعمال هر گونه تغییر در ان)

۲۸- تعبیه شدن کلیه سوکت ها در قالب IDC ها ۱۰ پایه و ساده تر شده ارتباط با انها

۲۹ - پورت مجرا جهت تغذیه دیگر امکانات جانبی (دارای ولتاژ های ۳,۳ و ۵ و ۱۲)

۳۰ - رایگان بدون نقشه ها :

برای راحتی شما ، ما نقشه تست شده هر دو برد را به صورت رایگان منتشر کرده ایم ، در صورتی که توانایی ساختن برد را دارید میتوانید نسبت به ساختن ان اقدام کنید .

برای دانلود نقشه ها به سایت کویر الکترونیک مراجعه کنید .

در بخش ضمائم به تشریح برد ها و نحوه کار با انها پرداخته شده است ، همچنین در اخرین صفحه مجله نحوه تهیه برد گفته شده است . جهت در یافت اطلاعات بیشتر به سایت مراجعه کنید .

برای شروع کار با arm سعی کرده ایم همه چیز را از ابتدا شروع کنیم ، ما اقدام به اموزش زبان c از پایه نموده ایم ، اشنایی با مفاهیم اولیه میکرو کنترلر ها (مفهموم بیت ، بایت ، پورت ، پایه ، حافظه ، و...) شما را در یادگیری این میکرو یاری میدهد . هر چند ما تمامی این موارد را در این شماره و شماره های بعدی بررسی خواهیم نمود .

مقدمه ای بر زبان c

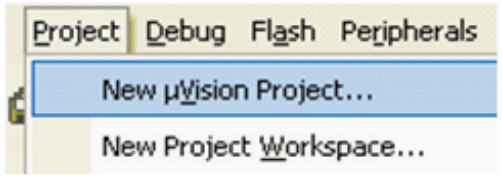
زبان برنامه نویسی C چیست؟

در سال ۱۹٦۷ زبان برنامه نویسی BCPL توسط martin richards طراحی و به جامعه جهانی معرفی شد. با توسعه و تکمیل این زبان طی سال های ۱۹٦۹ تا ۱۹۷۰ توسط ken thompson زبان برنامه نویسی دیگری به نام b متولد شد ، بعد از گذشت ۲ سال از ایجاد زبان c تحولات بسیاری در ان توسط برنامه نویسان و... ایجاد شد و کلیه خطا ها و باگ های ان برطرف گردید و در نهایت در ۱۹۷۱ این زبان با نام جدید C به دنیا معرفی شد .در علم الکترونیک و کار با میکرو ها ، کامپایلر ها و مفسرهای زیادی برای این زبان وجود دارد ، این کامپایلر ها کد های نوشته شده به زبان C را به اسمبلی و سپس به HEX تبدیل میکنند . تبدیل کد C به اسمبلی توسط برنامه کامپایلر (مانند CODE VSION یا IAR یا KIEL یا ...) انجام میشود و تبدیل برنامه اسمبلی به کد هگز بر عهده اسمبلر ارائه شده توسط سازنده میکرو کنترلر است . قبل از انجام هر کاری به بررسی محیط کامپایلر KEIL می پردازیم :

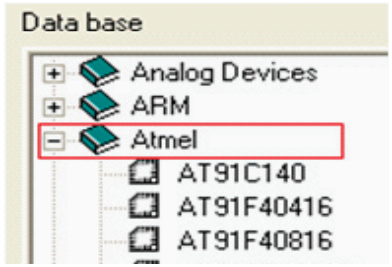
اموزش نصب و کرک این نرم افزار در قسمت ضمائم اورده شده است ، شما میتوانید با مراجعه به صفحه ۵۵ این اموزش را مشاهده کنید .

بعد از اینکه نرم افزار را نصب کردید ان را باز کنید :

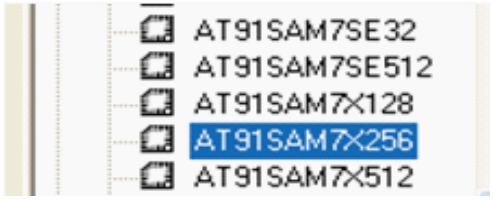
برای نوشتن برنامه به زبان c ابتدا باید یک پروژه ایجاد کنید ، به منوی project برید و در انجا گزینه ی new uvision project را انتخاب کنید :



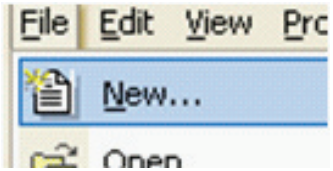
در پنجره ای که باز میشود یک نام مناسب برای پروژه وارد کنید و ان را در مسیر دلخواه ذخیره کنید .
بعد از انجام عملیات ذخیره سازی پنجره ای باز میشود ، در این پنجره شما باید پرادزنده مورد نظر خود را انتخاب کنید (پردازنده ای که میخواهید برایش برنامه بنویسید) جهت هماهنگی با مطالب کتاب از شاخه ی atmel :



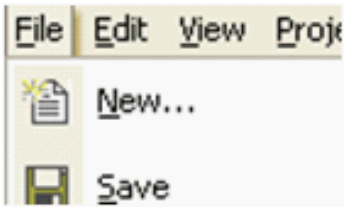
پردازنده ی AT۹۱SAMvX۲۵٦ را انتخاب کنید ، (این میکرو ، پردازنده برد موجود میباشد) و سپس بر روی ok کلیک کنید ، پیغامی که ظاهر میشود را نیز تایید کنید .



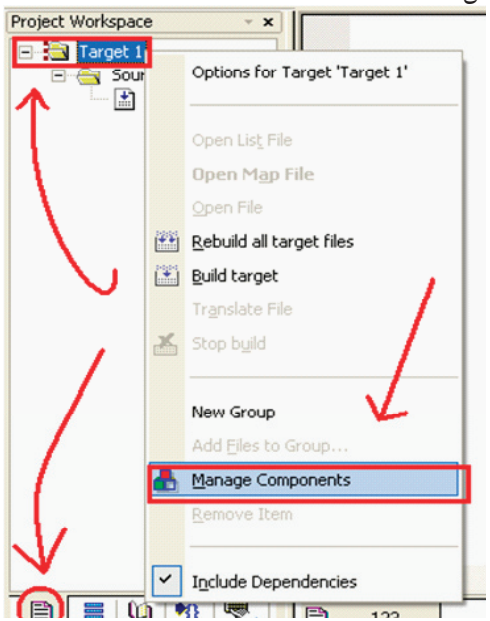
از منوی file گزینه ی new را انتخاب کنید :



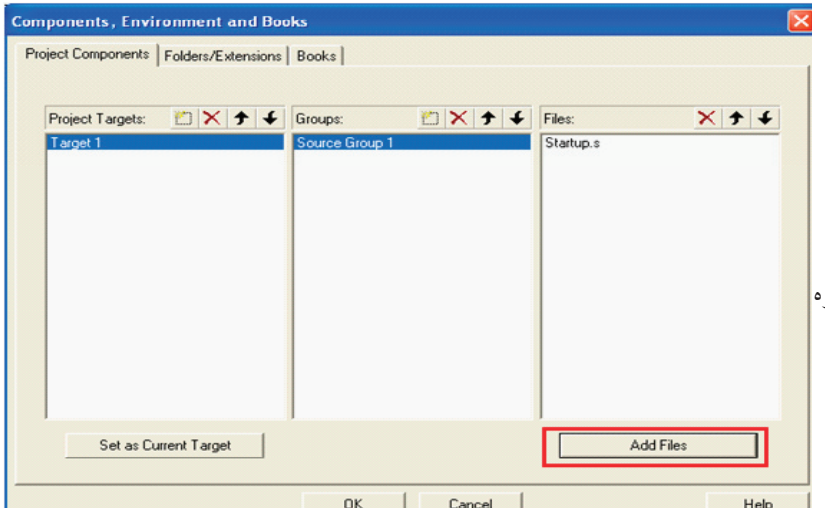
مشاهده میکنید که یک ویرایش گر متن در صفحه باز میشود ، از منوی فایل گزینه ی save را انتخاب کنید و فایل را در کنار پروژه با نام دلخواه و با پسوند .c (برای درج پسوند در اخر نام عبارت .c را بنوسید)ذخیره کنید .



اکنون باید فایل متنی را به پروژه معرفی کنید برای اینکار در پالت project workspace روی گزینه ی target ۱ کلیک راست کنید و در انجا گزینه ی manege components را انتخاب کنید . در صورتی که پالت project workspace در برنامه شما وجود ندارد از منوی view گزینه ی project window را انتخاب نمایید ، همچنین دقت کنید که گزینه ی file انتخاب شده باشد (گزینه ای که در پایین پالت با فلش مشخص شده)



بعد از انتخاب manege components پنجره ی زیر باز میشود :

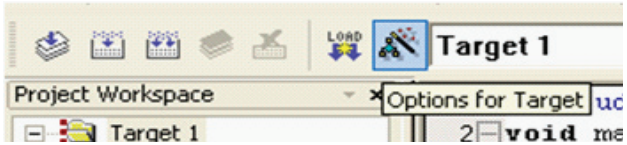


بر روی add file کلیک کنید و در پنجره ای که باز میشود ، فایل متنی که با پسوند .c ذخیره کردید باز کنید (بر روی add یکبار کلیک کنید و سپس پنجره را ببندید)
مشاهده میکنید که با کلیک روی ok فایل متنی به project workspace افزود ه میشود .
مراحل ایجاد پروژه به پایان رسید .

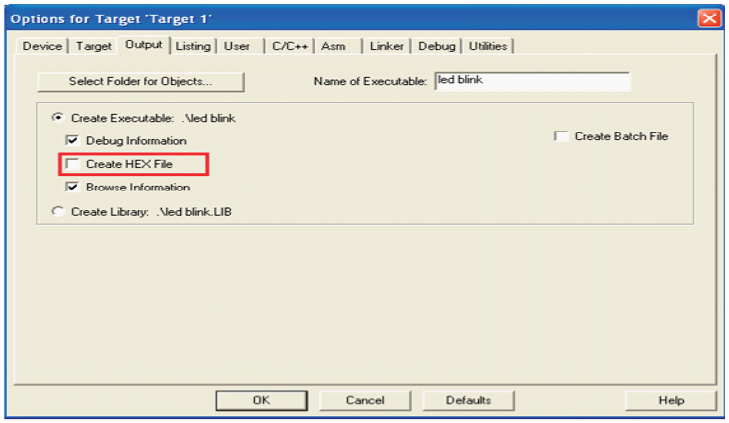
برای شروع برنامه زیر را در فایل متنی که از این به بعد به ان ویرایشگر گفته میشود کپی کنید . (از سخت بودن این کد ها نترسید ، وقتی که انها را یادبگیرید به اسانی انها پی خواهید برد)

```
#include <AT۹۱SAMvX۲۵۶.H>                /* AT۹۱SAMTvX۲۵۶ definitions */
#include <lib_AT۹۱SAMvX۲۵۶.h>
void wait(void);
int main (void) {
 AT۹۱F_PIO_CfgOutput(AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۰);
    AT۹۱F_PIO_ClearOutput(AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۰);
    wait();
    AT۹۱F_PIO_SetOutput  (AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۰);
    wait();
}
void wait (void) {
 unsigned int n;
 for (n = ۰; n < ۷۳۷۲۸۰۰; n++);
}
```

در بالای پنجره ی project workspace و بر روی ایکون options for target کلیک کنید یا از منوی flash گزینه ی Configure Flash tools را انتخاب نمایید

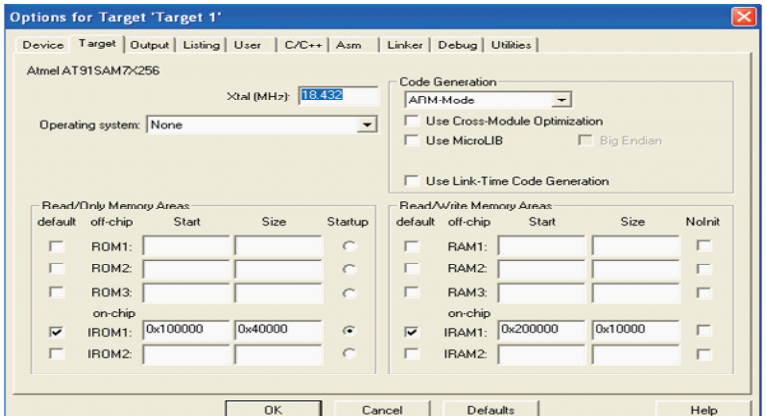


در پنجره باز شده ، تب output را انتخاب کنید و گزینه ی greate hex file را تیک بزنید و سپس بر روی ok کلیک کنید .



با انجام دادن عمل بالا فایل هگز به خروجی افزوده میشود .

در همین پنجره (پنجره بالا) بر روی target کلیک کنید و در بخش xtal{mhz} مقدار فرکانس کاری میکرو را مشخص کنید (مقدار کریستالی که به میکرو متصل است را در این بخش بنویسید .)

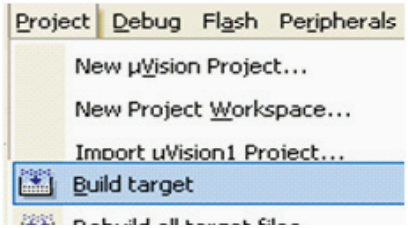


در صورتی که از برد اموزشی ما استفاده میکنید ، مقدار کریستال ۱۸،۴۳۲ مگاهرتز است .

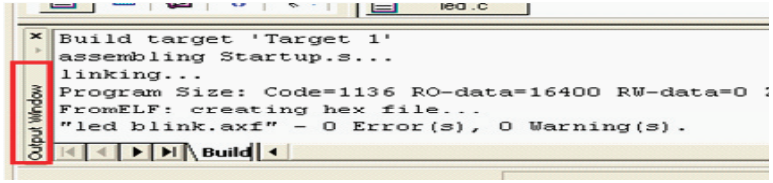
نکته :

مقدار کریستال ، با زمان تولید شده رابطه مستقیم دارد ، در صورتی که مقدار کریستال نوشته شده در این مکان ، با کریستال متصل شده به میکرو یکی نباشد ، برنامه به درستی اجرا نمیشود .

در این مرحله قصد کمپایل کردن برنامه را داریم ، برای اینکار به منوی Project بر وید و در انجا گزینه ی build target را انتخاب کنید :

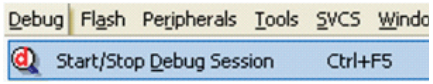


با این کار برنامه کامپایل میشود و کد هگز مربوطه در محل ذخیره فایل اصلی ذخیره میشود . در صورتی که خطا یا اشکالی در برنامه وجود داشته باشد ،در قسمت output window پیغام خطا به نمایش در میاید .(در صورتی که پالت output window در برنامه شما وجود ندارد از منوی view گزینه ی output window را انتخاب نمایید

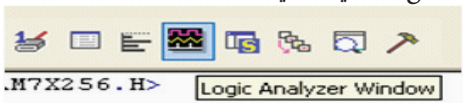


مراحل ذکر شده ، در تمامی پروژه باید انجام شود . اکنون وقت ان رسیده که برنامه خود را ازمایش کنیم و سپس ان را به میکرو انتقال دهیم .....

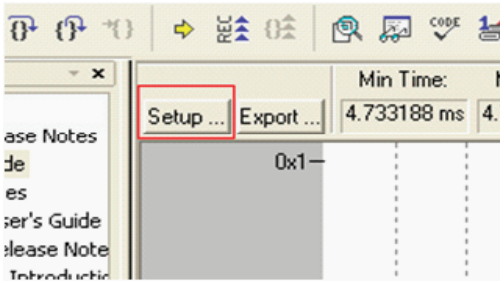
در این نرم افزار امکان شبیه سازی برنامه نوشته نیز وجود دارد ، برای ورود به محیط شبیه سازی ، بعد از کامپایل کردن برنامه از منوی debug گزینه ی start / stop debug session را انتخاب کنید :



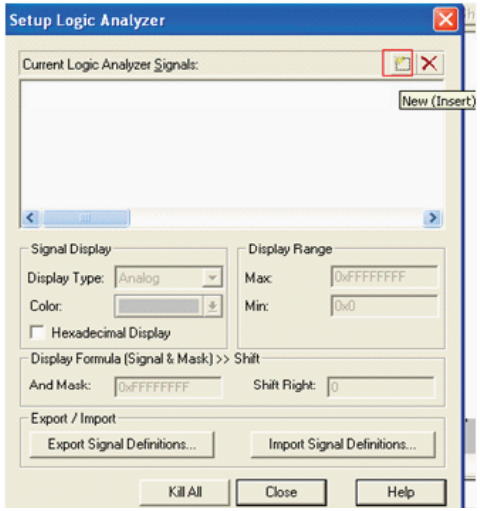
پنجره شبیه سازی بازمیشود ، در تولبار بر روی گزینه ی logic analyzer کلیک کنید :

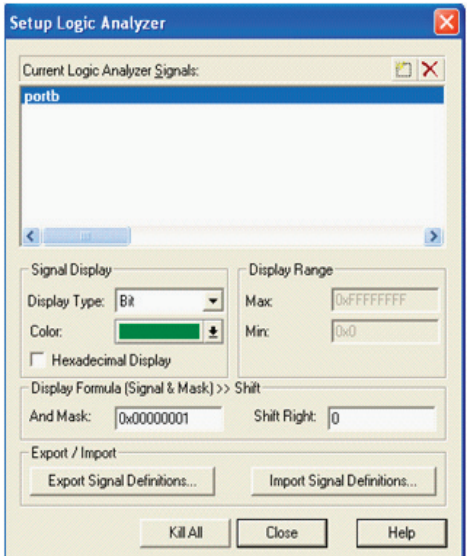


در پنجره باز شده بر روی setup کلیک کنید :



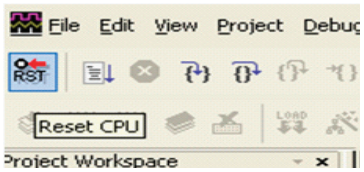
در پنجره باز شده ، بر وی new کلیک نمایید و در بخش ایجاد شده ، پورت که قصد شبیه سازی ان را داریم وارد کنید (portb)



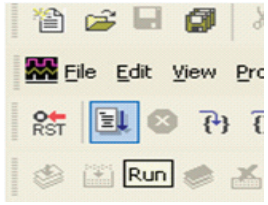


در همین پنجره ، بعد از وارد کردن نام پورت ، در بخش display type خاصیت را به bit تغییر دهید ، همچنین در بخش and mask ادرس پایه مورد نظر را وارد کنید (پایه شماره ۰ به ادرس ۰x۰۰۰۰۰۰۰۱)

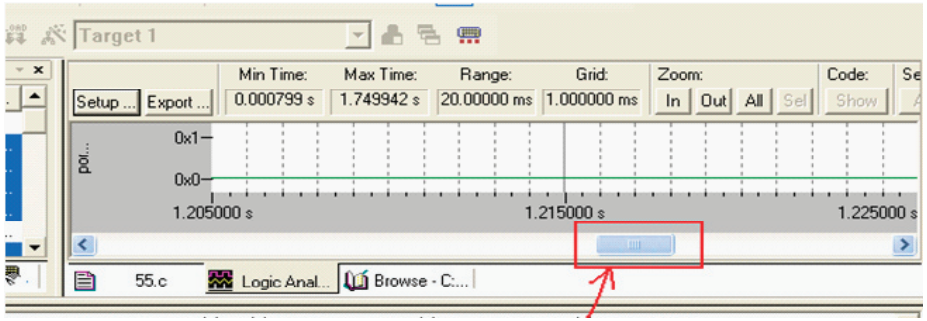
پنجره بالا را ببندید و در تولبار بر روی reset cpu کلیک کنید



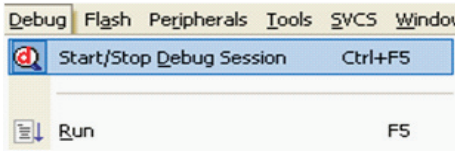
در تولبار بر روی run کلیک کنید و در صفحه اصلی عمل کرد برنامه را ببنید



مشاهده میکنید که ما یک موج مربعی با زمان تناوب ۱,۲۳ ثانیه اجاد کرده ایم ، در صورتی که موقعیت را تغییر دهید ، میتوانید بهتر شکل موج را ببنید



شما موفق به کامپایل و اجرا یک برنامه به زبان c شدید ، در شماره های بعدی به بررسی دیگر بخش های این نرم افزار خواهیم پرداخت . برای خروج از محیط شبیه سازی از منوی debug گزینه ی start / stop debog session را انتخاب کنید.



اکنون باید کد هگز تولید شده را به میکرومنتقل کنیم ، برای انتقال کد هگز به میکرو یا پروگرام کردن ان روش های مختلفی وجود دارد ، در ادامه به بررسی این روش ها پرداخته ایم ، اما قبل از خواندن انها توجه شما را به خواندن نکاتی پیرامون راه اندازی میکرو ذکر شده جلب میکنیم :

۱–در صورتی که از برد ارائه شده توسط گروه ما استفاده میکنید ، حتما راهنمای برد را مطالعه کنید . (راهنما در ضمیه ها اورده شده است)

۲–برای اتصال میکرو به پورت usb از کابل های اماده (استاندارد) استفاده کنید .

۳–کلیه مواردی که در ادامه ذکر شده اند (طول کابل ها، شماره قطعات و...) بر اساس مطالب موجود در سایت شرکت اتمل میباشد ، رعایت نکردن این موارد ممکن است باعث اسیب رسیدن به میکرو شود .

٤–در هنگام پروگرام کردن میکرو ، سخت افزار های جانبی را از ان جدا کنید .

روش های پروگرام کردن میکرو :

برای انتقال برنامه از کامپیوتر به میکرو روش های مختلفی وجود دارد ، این روش ها عبارتند از :

Serial Fast Flash Programming(IEEE® ۱۱٤۹.۱ JTAG)

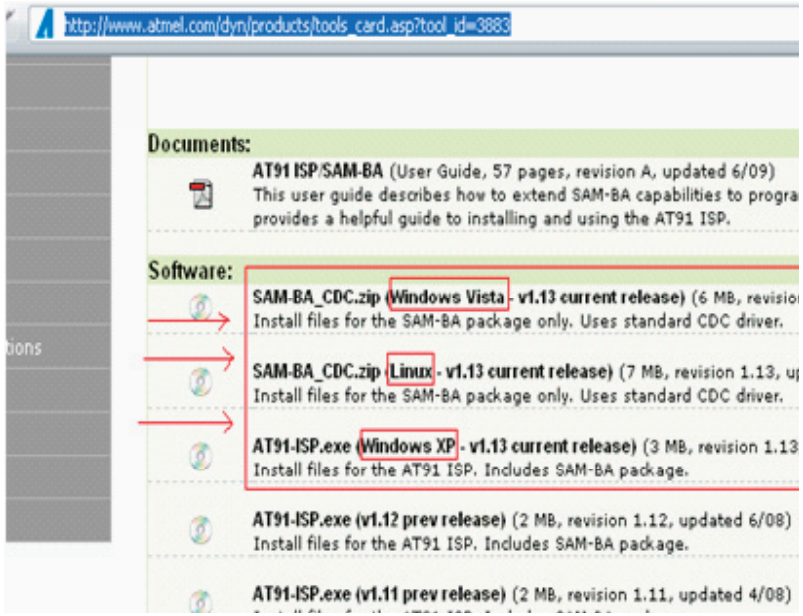
SAM-BA ® Boot

Parallel Fast Flash Programming

برای روش اول به سخت افزار(پروگرامر) jtag و برای مورد سوم به سخت افزار (پروگرامر) ppi (Parallel Programming Interface) نیاز دارید . مورد دوم تقریبا به سخت افزار جانبی نیاز ندارد .

برای راه اندازی این قابلیت به نرم افزار AT۹۱-ISP نیاز دارید ، این نرم افزار به صورت رایگان و برای انواع سیستم عامل ها ، از طرف شرکت اتمل ارائه شده است ، به ادرس زیر مراجعه کنید و نرم افزار مناسب را دانلود نمایید (با توجه به نوع ویندوز و سیستم عامل)

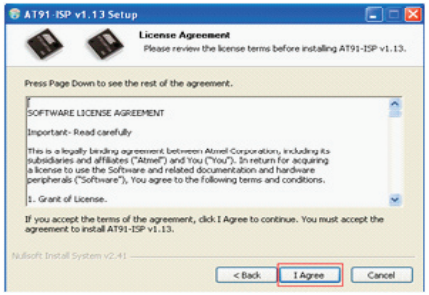
http://www.atmel.com/dyn/products/tools_card.asp?tool_id=۳۸۸۳



نصب نرم افزار SAM-PROG و sam-ba

بعد از انجام دانلود فایل Install AT۹۱-ISP v۱.۱۳.exe را اجرا کنید :

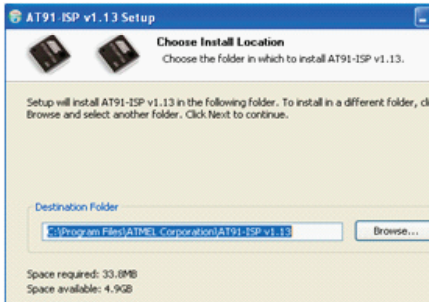
در اولین پنجره باز شده بر روی next کلیک کنید :



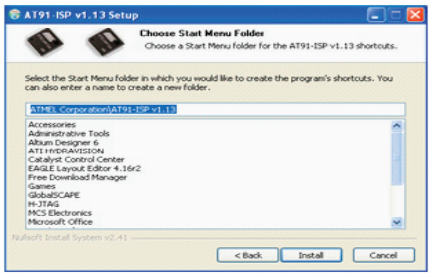
در پنجره دوم با قرداد موافقت نمایید :

در پنجره سوم بر روی next کلیک کنید :

در پنجره چهارم ، در صورت نیاز محل نصب نرم افزار را تغییر دهید ، و سپس بر روی next کلیک نمایید :



در پنجره پنجم بر رویinstall کلیک کنید :

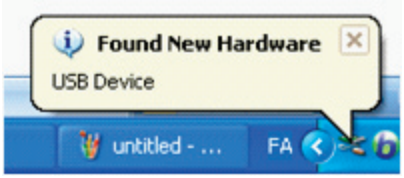


اندکی صبر کنید تا فایل ها کپی شوند ، سپس گزینه ی next را انتخاب کنید ، در پنجره های بعدی نیز ، next را انتخاب نمایید و در اخرین پنجره گزینه ی finish را انتخاب کنید :

مراحل نصب نرم افزار AT۹۱-ISP l یا sam-ba software به پایان رسید .

اکنون برد را به پورت usb متصل کنید ، برای اینکار کافی است ، کابل usb را به کامپیوتر و برد متصل کنید و سپس جامپر on/off را متصل نمایید.

در اولین اتصال برد پیغام روبرو نمایش داده میشود :



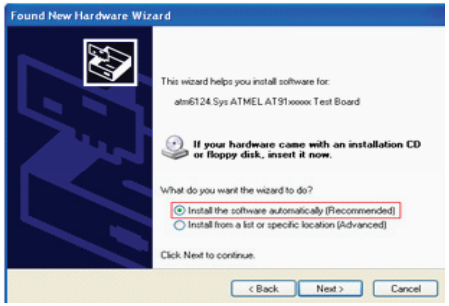
اندکی صبر کنید تا شناسی سخت افزار به پایان برسد :

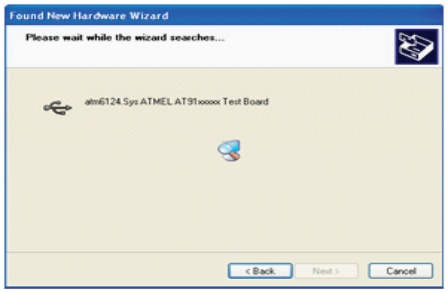
بعد از گذشت چند ثانیه پنجره found new hardware wizard باز میشود . در صورتی که این پنجره باز نشد به کنترل پانل بروید و گزینه ی add hardware را انتخاب نمایید (در صورتی که پیغام های بالا به نمایش در نیامد ، به قست های بعدی مراجعه کنید).

در پنجره found new hardware wizard گزینه ی yes, this time only را نتخاب کنید و سپس بر

روی next کلیک کنید :

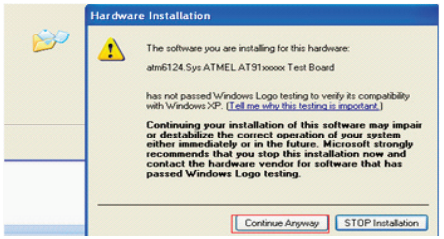
در پنجره بعدی نیز بر روی next کلیک نمایید :

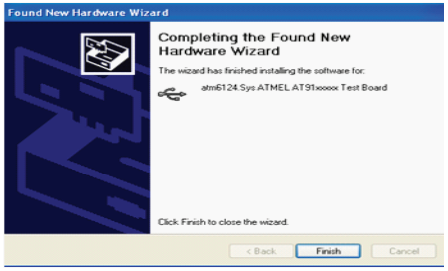




کمی صبر کنید تا سیستم درایو مناسب را برای سخت افزار متصل شده پیدا کند :

بعد از گذشت چند ثانیه پنجره ای باز میشود و از شما در مورد کپی کردن درایور سوال میکند ، در ان پنجره گزینه ی continue anyway را انتخاب نمایید :



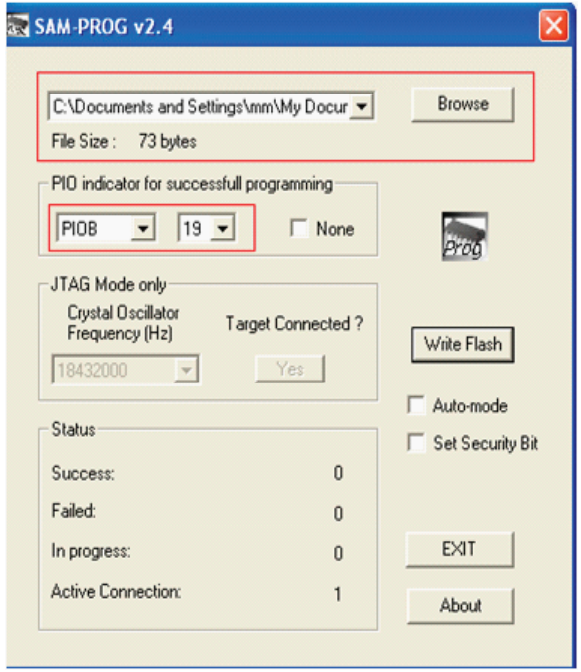


سخت افزار جدید با موفقیت نصب شد ، بر روی finish کلیک کنید تا پروسه تمام شود :

اکنون از مسیر Start Menu\Programs\ATMEL Corporation\AT٩١-ISP v١.١٣ گزینه ی SAM-PROG v٢.٤ را انتخاب کنید .

پنجره SAM-PROG باز میشود ، برای اطمینان از صحت مراحل انجام شده بر روی browse کلیک کنید و فایل tst.bin را در پوشه پیوست انتخاب نمایید .

در بخش pio مطابق شکل پایه b.١٩ را انتخاب کنید و سپس گزینه ی write flash را بزنید . در صورتی که تنظیمات به درستی انجام شده باشند ، led موجود بر روی برد خاموش میشود (پایه b.١٩ در سطح منطقی صفر قرار میگیرد)

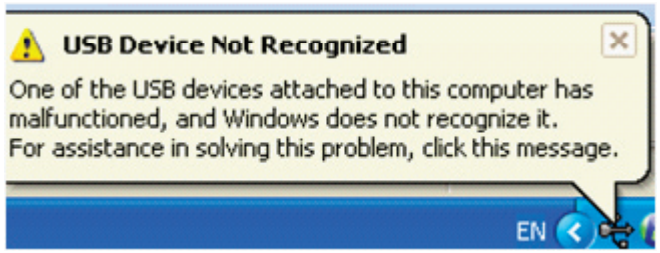


نکته ها

برای استفاده از sam-ba ، باید کریستال ۱۸.۴۳۲ مگا هرتز به میکرو متصل کنید (sam-ba فقط با این کریستال کار میکند.) در برد آموزشی ما این کریستال متصل می باشد

برای استفاده از sam-ba نیاز به انجام هیچ گونه تنظیماتی در ابتدای کار نیست (در صورتی که سخت افزار درست باشد ، با اولین اتصال دستگاه شناخته میشود .)

توجه داشته باشید که باید برد را مانند انوع سخت افزار های usb ، همچون فلش مموری و... متوقف کنید و سپس ان را از پورت بیرون بکشید



در صورتی که در هنگام اتصال دستگاه با پیغام زیر روبرو شدید موارد زیر را انجام دهید :

تغذیه برد را متصل کنید .جامپر پایه erase را متصل نمایید .تغذیه برد را قطع کنید .جامپر erase را بردارید .

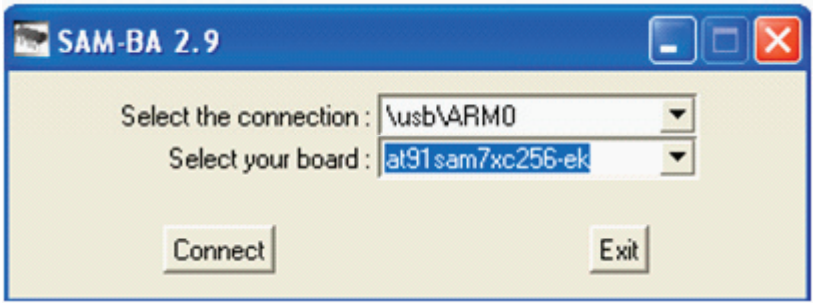
در برد آموزشی این پایه به اسم ERS می باشد که در ضمیمه کاملا توضیح داده شده است

اکنون برد را به usb متصل کنید ، با این کار مشکل حل میشود ، در صورتی که مشکل حل نشد ، ممکن است نرم افزار دانلود شده متناسب با ویندوز شما نباشد . یا ویندوز شما دچار اشکال باشد . (در انجمن مطرح کنید).

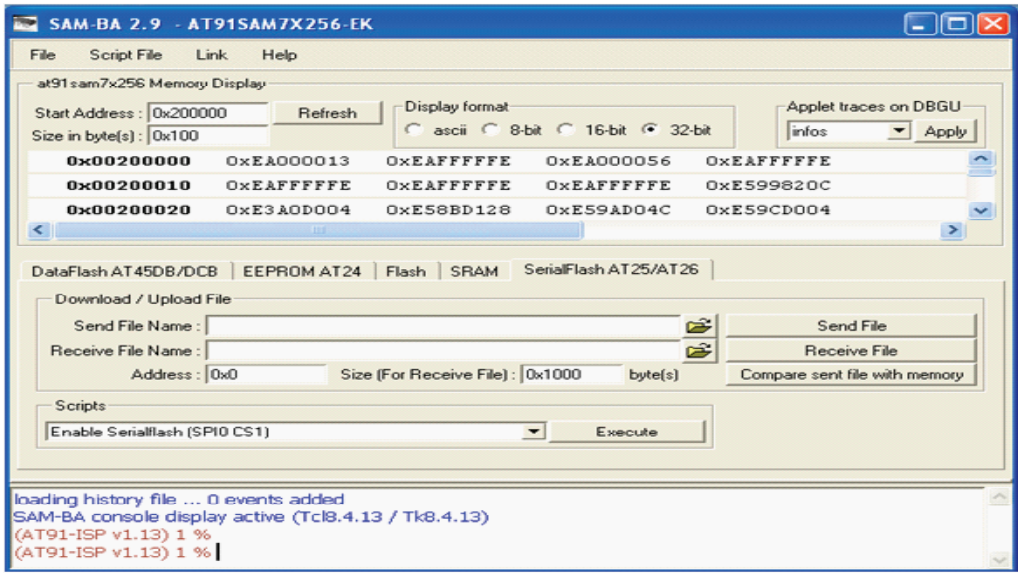
امکانات بیشتر

از مسیر Start Menu\Programs\ATMEL Corporation\AT۹۱-ISP v۱.۱۳ گزینه ی SAM-BA v۲.۹ را انتخاب کنید .

در پنجره باز شده و در بخش select your board میکرو متصل شده به پورت usb را انتخاب نمایید (گزینه ی at۹۱sam۷x۲۵۶-ek را انتخاب کنید) و سپس بر روی connect کلیک کنید :



مشاهده میکنید که پنجره sam-ba باز میشود . در این نرم افزار امکانات بیشتری در اختیار شما قرار دارد ، شما میتوانید حافظه های میکرو را برنامه ریزی کنید ، یا انها را بخوانید :



با نرم افزار SAM-PROG شما میتوانید به راحتی کد هگز یا باینری را از کامپیوتر خود به حافظه فلش میکرو منتقل کنید . همچنین نرم افزار SAM-ba امکان خواندن و نوشتن حافظه eeprom ، flash و... را به شما میدهد ، در اموزش برنامه نویسی ، شما را با این دو نرم افزار بیشتر اشنا خواهیم کرد ، بنا براین عجولانه عمل نکنید ، چون ممکن است به میکرو و برد خود اسیب بزنید .

همانطور که میدانید خروجی اکثر کامپایلر ها کد هگز میباشد ، توسط SAM-PROG شما میتوانید فقط کد باینری را بر روی میکرو بریزید (.bin) . برای تبدیل کد هگز به باینری از نرم افزار رایگان H-Converter استفاده میکنیم . این نرم افزار به همراه نرم افزار HJTAG نصب میشود . در ادامه نحوه استفاده از ان اورده شده است .

JTAG IEEE® ۱۱۴۹.۱ (Serial Fast Flash Programming)

Test Access Portand Boundary-Scan Architecture ۱۹۹۰-۱۱۴۹.۱ IEEE Standard یا jtag یک پروتکل ارتباطی میباشد که توسط تعدادی از شرکت های وابسته به سازمان ieee و تحت استاندارد ان به ثبت رسیده است .

در پروتکل jtag دسترسی کامل به cpu و حافظه ها فراهم میباشد ، و شما میتوانید داده های پردازش شده یا در حال پردازش توسط انها را مشاهده کنید ، شما میتوانید داده را در حافظه ها بنویسید یا از انها بخوانید . در این پروتکل ، امکانات زیادی در اختیار شما گذاشته شده است .

رابط jtag از ٤ پایه اصلی برای ارتباط با سخت افزار استفاده میکند :

هر وسیله ای که با استاندارد Jtag سازگار باشد لازم است تا پین های زیر را داشته باشد :

۱- Clock : (Test Clock Input) TCK ورودی ، این پالس برای همزمانی میان دستگاه مورد تست و پروگرامر jtag میباشد .

۲- TDI (Test Data In) : از طریق این پایه ، داده از پروگرامر به دستگاه در حال تست وارد میشود .

۳- TDO (Test Data Out): از طریق این پایه ، داده از دستگاه مورد تست به پروگرامر میرود .

٤- TMS (Test Mode Select) : از طریق این پورت حالتهای مختلف تست انتخاب می شود.

همچنین در این میان دو پایه دیگر نیز وجود دارد :

۵- TRST (Test Reset Input) این پایه از پروگرامر به ریست دستگاه متصل میشود و قبل از شروع کار ان را باز نشانی میکند .

٦- JTAGSEL (JTAG SELECT) : برای راه اندازی پروتکل JTAG ، این پایه باید یک شود .

برای کار با JTAG به موارد زیر نیاز دارید :

۱- کابل ارتباطی – سخت افزار JTAG – نرم افزار JTAG

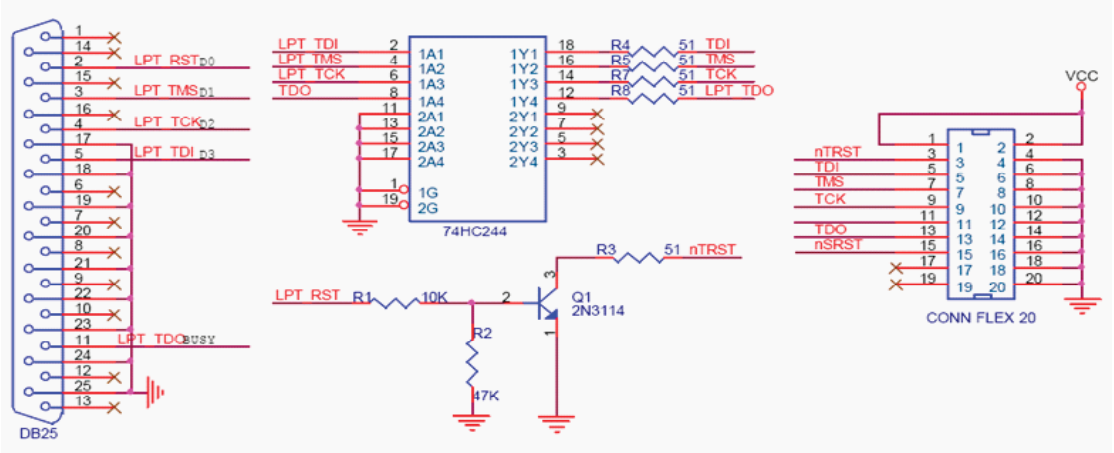
کابل ارتباطی وظیفه اتصال دستگاه مورد تست را به پروگرامر را به عهده دارد ، معمولا پروگرامر از طریق یک کابل دیگر به پورت سریال یا موازی کامپیوتر متصل میشود .سخت افزار JTAG وظیفه کنترل داده های ارسالی و دریافتی و مراقبت از کامپیوتر در برابر خطا های شخصی را به عهده دارد و وظیفه نرم افزار JTAG ، تبدیل کد های دریافتی به علائم نمایشی و کد های قابل فهم برای انسان میباشد .

ما برای راه اندازی واسط jtag از نرم افزار رایگان HJTAG استفاده میکنیم ، شما میتوانید این نرم افزار را از لینک زیر در یافت کنید :

http://www.hjtag.com/download.html

نصب نرم افزار تقریبا مانند نصب نرم افزار AT۹۱–ISP میباشد .و نکته خاصی ندارد .

برای اتصال میکرو به کامپیوتر به یک سخت افزار نیاز دارید ، این سخت افزار مطابق شکل زیر است :



کانکتور تعبیه شده برای اتصال کابل jtag باید مشابه کانکتور تعبیه شده بر روی برد باشد، تا در اتصال سیم ها مشکلی بوجود نیاید .

در صورتی که پایه هیا پورت lpt را روبروی خود بگیرید ، شماره پایه ها در کنار انها نوشته شده است .

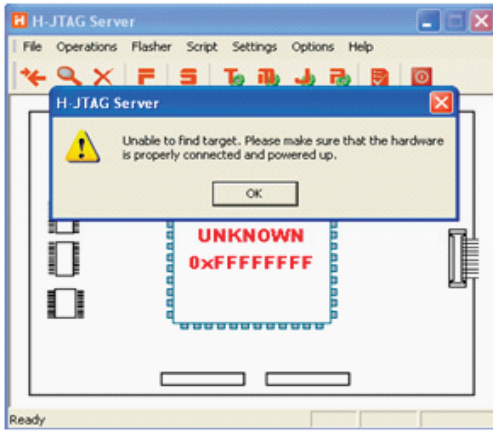
به دلیل اختلاف منطقی در سطح پالس ریست ، استفاده از ترانزیستور و مقاومت ها الزامی است .در پوشه ضمیمه ، سند شماتیک و pcb پروگرامر بالا وجود دارد (فایل پروتل).

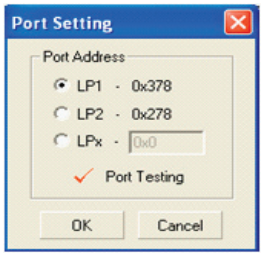
بعد از نصب نرم افزار و ساختن پروگرامر ، ان را به برد متصل نمایید . شما باید جامپر jselect را نیز متصل کنید .

پروگرامر را به پورت lpt متصل کنید و سپس نرم افزار را باز نمایید .

در صورتی که سخت افزار پروگرامر درست باشد ، در پنجره hjtag نام هسته استفاده شده در میکرو به نمایش در میاید :

در صورتی که با پیغام مقابل روبرو شدید :





به منوی settings بروید و بعد از انتخاب گزینه ی port setting در پنجره باز شده اولین مورد را انتخاب کنید (در صورتی که کامپیوتر شما بیشتر از یک پروت دارد ، گزینه هیا بعدی را انتخاب نمایید)

بعد از انجام تنظیمات از منوی felasher گزینه ی start h-flasher را انتخاب کنید .

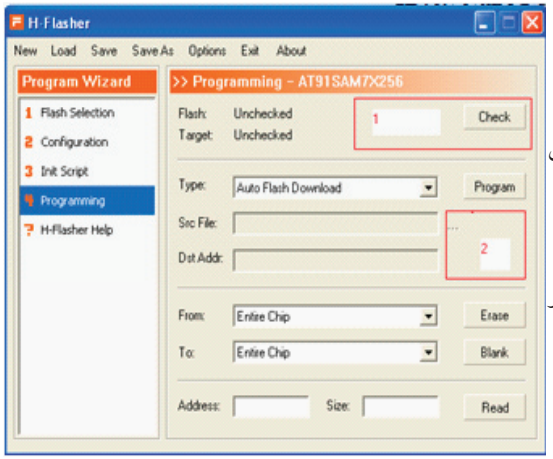
ابتدا بر روی check کلیک کنید تا میکرو شناسایی شود ، سپس در محل دو بر روی .... کلیک کنید و فایل با پسوند .hex یا .bin را انتخاب کنید .

با زدن گزینه ی program برنامه از کامپیوتر به میکرو منتقل میشود .

در اینده با نحوه دیباگ کردن برنامه با این نرم افزار اشنا خواهیم شد .

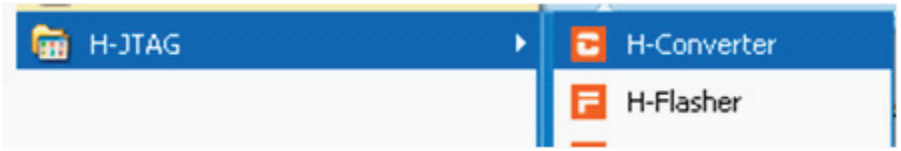
نحوه تبدیل کد هگز به باینری

برای تبدیل کد هگز به باینری از نرم افزار H-Converter مجموعه hjtag استفاده میکنیم .نرم افزار hjtag را نصب کنید و سپس گزینه ی H-Converter را انتخاب نمایید :

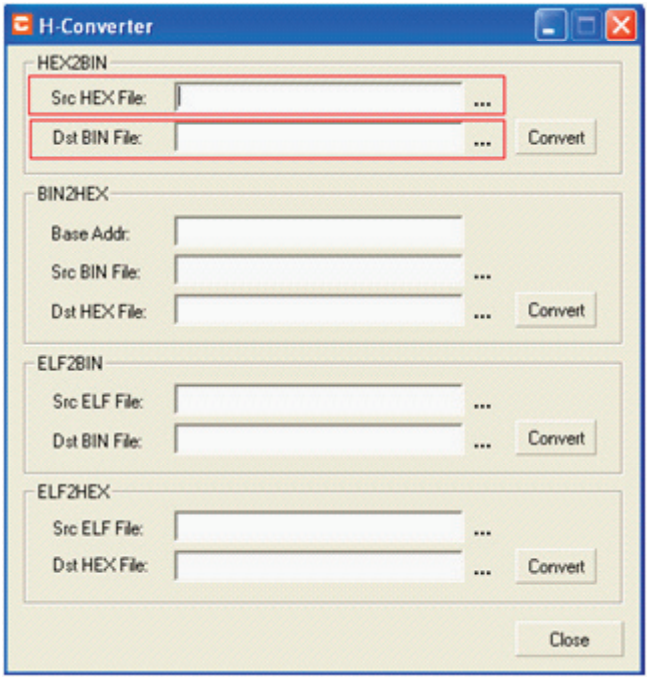


در بخش hex۲bin و در قسمت src hex file بر روی .... کلیک کنید و کد هگز را باز کنید . نرم افزار به صورت پیش فرض کد باینری را در کنار کد هگز ذخیره میکند .

بر روی convert کلیک کنید تا عملیات تبدیل اغاز شود . بعد از اتمام تبدیل با پیغام نرم افزار مبنی بر موفقیت تبدیل روبرو خواهید شد .



در این نرم افزار امکان تبدیل سایر کد ها به یکدیگر نیز وجود دارد .
همانطور که قبلا گفتیم رابط SAM-BA کد باینری را به میکرو منتقل میکند و خروجی نرم افزار KEIL کد هگز میباشد . برای تبدیل کد هگز خروجی KEIL مراحل بالا را انجا دهید (کد هگز در محل ذخیره برنامه ایجاد شده است) و سپس ان را به میکرو منتقل کنید ،شما باید به پایه b.۰ یک led متصل کنید (led با مقاومت ۱۵۰ اهم سری شده باشد)
در برد آموزشی این LED به رنگ زرد می باشد و با نام LED
بعد از پروگرام کردن برنامه کلید ریست را فشار دهید ، مشاهده میکنید که led شروع به چشمک زدن میکند . برنامه زیر را کامپایل کنید و بر روی میکرو بریزید ، در این برنامه پایه b.۱۹ شروع به چشمک زدن خواهد کرد (led روی برد)




```
#include <AT۹۱SAMvX۲۵۶.H>                /* AT۹۱SAMTvX۲۵۶ definitions */
#include <lib_AT۹۱SAMvX۲۵۶.h>
void wait(void);
int main (void) {
 AT۹۱F_PIO_CfgOutput(AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۱۹);
    AT۹۱F_PIO_ClearOutput(AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۱۹);
    wait();
    AT۹۱F_PIO_SetOutput  (AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۱۹);
    wait();
}
void wait (void) {
 unsigned int n;
 for (n = ۰; n < ۷۳۷۲۸۰۰; n++);
}
```

اموزش زبان c

بلوگ دیاگرام یک برنامه به زبان c تقریبا به شکل زیر است :

فراخوانی و راه اندازی پردازنده و کتابخانه ها و فایل های هدر و....

پیکربندی امکانات (مانند lcd و...)

معرفی متغیر ها
شروع حلقه
برنامه ای که باید انجام شود
پایان حلقه
پایان برنامه
زیر برنامه ها (در صورت وجود)

از سخت بودن این کد ها نترسید ، وقتی با انها کار کنید می فهمید که هر یک چه معنی دارد .

```
#include <AT۹۱SAMvX۲۵٦.H>
#include <lib_AT۹۱SAMvX۲۵٦.h>
void wait(void);
int main (void) {
  AT۹۱F_PIO_CfgOutput(AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۰);
     AT۹۱F_PIO_ClearOutput(AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۰);
     wait();
     AT۹۱F_PIO_SetOutput  (AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۰);
     wait();
}
void wait (void) {
  unsigned int n;
  for (n = ۰; n < ۷۳۷۲۸۰۰; n++);
}
```

```
#include <AT۹۱SAMvX۲۵٦.H>
```

این دستور نوع میکرو را مشخص میکند ، شما میتوانید هر یک از میکرو های موجود را انتخاب کنید .

```
#include <lib_AT۹۱SAMvX۲۵٦.h>
```

با این دستور ، فایل هدر h.lib_AT۹۱SAM۷X۲۵۶ را به برنامه فرا خوانی میکنیم ، در ادامه با فایل هدر اشنا خواهیم شد .

```
void wait(void);
```

این دستور مشخص کننده وجود یک زیر برنامه است .

```
void main (void){
```

این بخش شروع حلقه ما خواهد بود ، cpu میکرو دستورات میان دو اکولاد را اجرا میکند و فقط با دستورات پرش یا شرط که در ادامه توضیح میدهیم از این حلقه بیرون میرود .

```
void  main() {
دستورها
}
```

در واقع اکولاد ها مشخص میکنند که دستورات مربوط به کدام حلقه میباشند ، در این بخش دستوات موجود مربوط به حقه ای به نام main است ، نام حلقه میتواند هر چیزی باشد

```
AT۹۱F_PIO_CfgOutput(AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۰);
```

با این دستور portb.۰ به عنوان خروجی تعریف میشود .

```
AT۹۱F_PIO_ClearOutput(AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۰);
```

این دستور ، مقادیر موجود بر روی پایه را پاک میکند (پایه مورد نظر را صفر میکند)

```
wait();
```

با این دستور زیر برنامه زیر را فراخوانی میکنیم :

```
void wait (void) {
  unsigned int n;
  for (n = ۰; n < ۷۳۷۲۸۰۰; n++);
}
```

در این زیر برنامه زمان تقریبی ۱,۲ ثانیه ساخته میشود و سپس به خط بعد از دستور wait پرش میشود .

```
AT۹۱F_PIO_SetOutput  (AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۰);
```

با این دستور پایه مورد نظر ۱ میشود .

در ادامه با کلیه دستورات اشنا خواهیم شد ، مشا هده کردید که تمامی برنامه ها یک قالب یکسان دارند و هر دستور در جای خودش میاید .

استفاده از توضیحات

گاهی اوقات در برنامه نیاز به توضیح دادن یک خط یا کد یا عمل است برای درج توضیحات از // استفاده میشود یا توضیحات میان /* .....*/ درج میشود .توضیحات توسط کامپایلر خوانده نمی شوند و شما میتوانید از انها برای توضیح برنامه استفاده نمایید .

معرفی متغیر ها

متغیر چیست ؟

متغیر ها مکانی از حافظه موقت هستند که به عمل خاصی اختصاص داده می شوند ، مثلا یک متغیر از نوع char که مقدار عدد روی یک پورت ۱۶ پایه ای درونش ریخته شده است ، ۱۶ بیت (خانه) از حافظه را اشغال میکند .

از متغیر ها برای ذخیره یک عدد یا وضعیت استفاده میشود ، از انجا که وضعیت های مختلف (از نظر اندازه) وجود دارند ، بنابراین به چندین نوع متغیر نیاز داریم ، keil متغیر های زیر را در اختیار برنامه نویس قرار میدهد :

Type	Size in bits	Natural alignment in bytes
char	8	1 (byte-aligned)
short	16	2 (halfword-aligned)
int	32	4 (word-aligned)
long	32	4 (word-aligned)
long long	64	8 (doubleword-aligned)
float	32	4 (word-aligned)
double	64	8 (doubleword-aligned)
long double	64	8 (doubleword-aligned)
All pointers	32	4 (word-aligned)
_Bool (C only[1])	8	1 (byte-aligned)

تعریف متغیر به فرم زیر است :

نام متغیر نوع متغیر ;

مثلا متغیر a از نوع int تعریف شده است و میتواند بین ۰ تا ۳۲ بیت را در خود ذخیره کنید (میتواند از ۰ تا ۴۲۹۴۹۶۷۲۹۶ ، تغییر کند) و متغیر b از جنس char میباشد و میتواند از ۰ تا ۸ بیت (از ۰ تا ۲۵۶) تغییر کند :

```
int a;
Char b;
```

توجه داشته باشید :

که نام متغیر ها نمی تواند شامل کلمات رزرو شده (دستورات که در برنامه نویسی به کار میروند مانند void و int و...) باشد ، همچنین نام متغیر نباید از ۳۱ کاراکتر (حرف) بیشتر باشد و زبان c بین حروف کوچک و بزرگ تفاوت قائل می شود (aبا A فرق دارد.)

دستورات و توابع ریاضی ومنطقی

در Keil دستوراتی برای انجام عملیات ریاضی وجود دارد ، این دستورات را در زیر مشاهده میکنید :

نوع عملیات	نما د	مثال	نتیجه	توضیح
ضرب	*	۲*۲	٤	
تقسیم	/	٦ /۳	۲	
جمع	+	٤+٤	۸	
تفریق	–	٥–۳	۲	متغیر نتیجه باید توانایی جذب عدد منفی را داشته باشد
افزایش یک واحد	++	++A	۱+A	در هر تکرار ، یک واحد به a اضافه میشود
کاهش یک واحد	--	--a	۱-a	در هر تکرار ، یک واحد از a کم میشود
قرینه	–	–(۱)	۱–	عدد در یک منفی ضرب میشود
بزرگتر	>	a>b	۲<۳	
کوچکتر	<	B<a	۳>۲	
کوچکترمساوی	=<	٦=>A	A=(٦ , ٥ ,...)	A میتوانید یکی از اعداد بزرگتر مساوی ٦ باشد
بزرگتر مساوی	>=	٦=<a	A=(٦,٧,...)	A میتوانید یکی از اعداد کوچکتر مساوی ٦ باشد
انتساب	=	A=b	٤=a,۱=B ۱=A=b , a	مقدار متغیر b در متغیر a ریخته میشود
تساوی	==	A= =b	۲=۲	دو متغیربا هم برابرند
باقیمانده تقسیم	%	A/B	۲=٤/۱۰	متغیر A به Bتقسیم میشود ،حاصل باقیمانده است، در مثال عدد ۲ برگردانده میشود
AND	&	A*B	۰x۰۰=۰x۰F & ۰xF۰	اگر بیت های همسان یک باشند،خروجی یک است در غیر اینصورت خروجی صفر میباشد
OR	|	A+B	۰x۰۳ =۰x۰۳ | ۰x۰۰	بیت های همسان با هم جمع می شوند و حاصل در خروجی ظاهر میشود . ۰x۰۳

XOR	^	۰xFF ^ ۰x۰F	۰xF۰	
مکمل یک	~	(۰Xf)~	۰x۰F	یکی از عملیات های زیاضی است
شیفت به راست	>>	٤<<a ۰xF۰=A	۰x۰F	مقدار اولیه ۱۱۱۱ در مبنای باینری است ، با این دستور مقدار اولیه به اندازه دلخواه به سمت راست حرکت داده میشود ، مقدار نهایی برابر ۱۱۱۱۰۰۰۰ در مبنایی باینری است
شیفت به چپ	<<	٤ >> ۰x۰F	۰xF۰	
نامساوی	=!	A!=b	۳ =!۲	
AND منطقی	&&	(۳=!۱)&&(۳<۲)	False	
OR منطقی	||	(۱۰) ||(۳>'a')	True	
نقض	!	!(۲<۵)	۵<۲	با این دستور ، از غلط بودن یک تساوی چشم پوشی میشود ، تساوی درست فرض میشود

.

شما میتوانید کلیه اعمال بالا را بر روی متغیر ها انجام دهید و حاصل را در یک متغیر ذخیره کنید .

دستورات حلقه و پرش

گاهی مواقع برنامه باید چندین بار تکرار شود ، برای اینکار از دستورات حلقه و پرش استفاده میشود ، در زیر به بررسی انواع حلقه های موجود در keil پرداخته میشود .

حلقه ی while :

: این حلقه به فرم زیر میباشد

```
while (x) {
برنامه
}
```

این حلقه که خود در حلقه ی اصلی قرار میگیرد ، یک حلقه ی بینهایت است ، هنگامی که برنامه به اکولاد دوم رسید به while پرش میکند ، شما میتوانید به جای x عدد دلخواه خود را بگذارید و از تعداد بیشتری حلقه در برنامه خود استفاده کنید .

حلقه ی for :

این حلقه ی یک حلقه ی شرطی است و تازمانی که شرط ان درست باشد ادامه می یابد ، این حلقه به فرم کلی زیر است :

```
For (مقدار اولیه حلقه , شرط پایان , گام حلقه){
برنامه
}
```

در این نوع حلقه میتوان تعداد تکرار را مشخص کرد ، مقدار اولیه حلقه ، میتواند یک عدد باشد ، شرط پایان حلقه نیز یک عدد یا متغیر است ، هنگامی که مقدار اولیه در شرطی که در این قسمت نوشته میشود ، صدق کند ، حلقه به پایان میرسد .

گام حلقه نیز مشخص میکند که مقدار اولیه چگونه تغییر کند :

```
for (j = ۱; j <۱۰ ; j++ ) {
برنامه
}
```

در مثال بالا مقدار اولیه j ۱ میباشد ، تا وقتی که j از ۱۰ کوچکتر باشد ، حلقه ادامه می یابد ، همچنین با هر بار تکرار حلقه یک واحد به j افزوده میشود .

حلقه ی do – while :

```
do
{
        برنامه
}while(شرط);
```

در این حلقه برخلاف دیگر حلقه ها ، شرط در انتهای حلقه چک میشود

این حلقه حداقل یکبار اجرا میشود ، در صورتی که در قسمت شرط دستور مبنی بر تکرار حلقه وجود داشته باشد ، حلقه دوباره تکرار خواهد شد .

دستور goto :

توسط این دستور میتوان به یک برچسب پرش کرد ، باز گشت از برچسب با دستور return انجام میشود :

```
Goto lable ;
Lable:
برنامه
Return;
```

برچسب باید در داخل حلقه ی اصلی باشد ، طول برچسب باید از ۳۱ کاراکتر کمتر باشد ، برای برچسب نمیتوان از کلمات رزرو شده استفاده کرد .

دستورات مربوط به پورت ها

در زبان c مجموعه از ریجستر ها وجود دارد که ما باید انها را بشانسیم ، شناخت این ریجستر ها برای کار با هر جزء میکرو الزامی است .

در کامپایلر keil هر یک از امکانات جانبی دارای ریجستر مخصوص به خود هستند ، مثلا ریجستر AT۹۱F_PIO مخصوص به پورت های میکرو کنترلر های اتمل میباشد ، شما میتوانید در این ریجستر مشخص کنید که پورت چه وضعتی داشته باشد ، شما میتوانید پورت یا پایه را به عنوان ورودی یا خروجی مشخص کنید و سپس مقادیر مختلف را در ان بنویسید یا از ان بخوانید .

در ادمه اموزش با کلیه ریجستر ها اشنا خواهیم شد .

فایل های هدر (کتابخانه) :

در هر برنامه فایل های با پسوند .h وجود دارند ، ما این فایل ها را با نام هدر می شناسیم . این فایل ها برای ساده تر کردن کد ها در هنگام برنامه نویسی به کمک ما می ایند مثلا فایل هدر lib_AT۹۱SAMvX۲۵۶.h دستورات مربوط به میکرو کنترلر at۹۱samvx۲۵۶ را به زبان گفتار نزدیک میکند ،. ما همچنین میتوانیم هدر یا کتابخانه دلخواه خود را بنویسیم . در keil هدر های زیادی وجود دارد که در اینده با انها اشنا خواهیم شد

ریجستر های مخصوص به پورت :

این دستورات و ررجیستر ها برای هدر های lib_AT۹۱SAM??????????.h گفته میشود ، برای اجرای این دستورات نیاز است که این کتابخانه ها را در برنامه خود فراخوانی کنید ، مثلا برای میکرو at۹۱samvx۲۵۶ :

```
#include <AT۹۱SAMvX۲۵۶.H>
#include <lib_AT۹۱SAMvX۲۵۶.h>
```

ادامه برنامه

یا میکرو AT۹۱SAMvS۶٤ :

```
<H.include <AT۹۱SAMvS۶٤#
#include <lib_AT۹۱SAMvS۶٤.h>
```

ادامه برنامه

قرار دادن پورت به عنوان ورودی یا خروجی :

```
AT۹۱F_PIO_CfgOutput(AT۹۱C_BASE_PIOx, AT۹۱C_PIO_Pxy);
```

دستور بالا پایه y از پورت x را به عنوان خروجی تعریف میکند .

```
(AT۹۱F_PIO_GetInput(AT۹۱C_BASE_PIOx) , AT۹۱C_PIO_Pxy)
```

دستور بالا پایه y از پورت x را به عنوان ورودی تعریف میکند .

```
AT۹۱F_PIO_ClearOutput(AT۹۱C_BASE_PIOx, AT۹۱C_PIO_Pxy)
```

دستور بالا ، پایه y از پورت x را صفر میکند .

```
AT۹۱F_PIO_SetOutput  (AT۹۱C_BASE_PIOx, AT۹۱C_PIO_Pxy)
```

دستور بالا پایه y از پورت x را صفر میکند .

مثال :

به portb.۰ میکرو کنترلر AT۹۱SAMvX۲۵۶ یک کلید و به portb.۱۹ یک led متصل کرده ایم ، برنامه ای بنویسید که با تحریک کلید led روشن و با قطع تحریک led خاموش شود :

```
#include <AT٩١SAMvX٢٥٦.H>
#include <lib_AT٩١SAMvX٢٥٦.h>
int main (void) {
  AT٩١F_PIO_CfgOutput(AT٩١C_BASE_PIOB, AT٩١C_PIO_PB١٩);
(AT٩١F_PIO_GetInput(AT٩١C_BASE_PIOB) , AT٩١C_PIO_PB٠) ;
  if (( AT٩١C_PIO_PB٠ == (٠)
  AT٩١F_PIO_SetOutput  (AT٩١C_BASE_PIOB, AT٩١C_PIO_PB١٩);
  else
 AT٩١F_PIO_ClearOutput(AT٩١C_BASE_PIOB, AT٩١C_PIO_PB١٩);
}
```

دستورات تاخیر

گاهی موقع لازم است ، برنامه برای مدتی اجرا نشود ، یرای این کار از دستورات تاخیر استفاده میشود ، در keil تابع برای تاخیر (در برنامه) وجود ندارد ، و شما باید برای ایجاد تاخیر cpu را به کار دیگری مشغول کنید ، یکی از این کار ها شمردن اعداد میباشد :

برنامه زیر را را شبیه سازی کنید:

```
 #include <AT٩١SAMvX٢٥٦.H>
#include <lib_AT٩١SAMvX٢٥٦.h>
void wait(void);
int n;
int main (void) {
  AT٩١F_PIO_CfgOutput(AT٩١C_BASE_PIOB, AT٩١C_PIO_PB١٩);
     AT٩١F_PIO_ClearOutput(AT٩١C_BASE_PIOB, AT٩١C_PIO_PB١٩);
     wait();
     AT٩١F_PIO_SetOutput  (AT٩١C_BASE_PIOB, AT٩١C_PIO_PB١٩);
     wait();
}
void wait (void) {
  for (n = ٠; n < ١٨٤٣٢٠٠; n++);
}
```

مشاده میکنید که در هر نیم میلی ثانیه پایه b۱۹ روشن و خاموش میشود .در این برنامه برای ایجاد تاخیر از فراخوانی توابع استفاده شده است ، در زیر برنامه را تحلیل میکنیم :

#H.include <AT۹۱SAMvX۲۵٦> در اولین خط برنامه میکرو کنترلر معرفی شده است ، میکرو مورد استفاده ما AT۹۱SAMvX۲۵٦ میباشد که قبلا در مورد نحوه انتخاب میکرو توضیح داده شد .

#h.include <lib_AT۹۱SAMvX۲۵٦> فراخوانی فایل هدر

void wait(void); در خط دوم مقداری از فضای حافظه برای توابع فراخوانی رزرو شده است .

int n;در خط سوم یک متغیر از نوع int معرفی شده است (میتواند از ۰ تا ۲بتوان ۳۲ ، یعنی ٤۲۹٤۹٦۷۲۹٦ تغییر کند).

void main(()) شروع حلقه ی اصلی برنامه میباشد .

AT۹۱F_PIO_CfgOutput(AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۱۹);

در این خط پایه b.۱۹ به عنوان خروجی پیکربندی شده است

AT۹۱F_PIO_ClearOutput(AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۱۹);

در این خط پایه b۱۹ در سطح منطقی صفر قرار گرفته است .

wait(); در خط هشتم تابع تاخیر فراخوانی میشود ، با رسیدن cpu به این خط ، به تابع void wait(void); پرش میشود ، در زیر برنامه wait یک حلقه ی for وجود دارد ، در این حلقه مقدار اولیه ی ۰ برای i در نظر گرفته شده است ، ، با

دستور i++ مدام به متغیر i یک واحد اضافه میشود ، هنگامی که i به ۱۸٤۳۲۰۰ رسید cpu به خط بعد از شرط پرش میکند ، در ان جا یک اکولاد وجود دارد که cpu را به حلقه ی اصلی برمیگرداند و cpu برنامه را از خط بعد از wait ادامه میدهد

AT۹۱F_PIO_SetOutput (AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۱۹);

در این خط پایه b۱۹ در سطح منطقی یک قرار گرفته است

wait(); در خط نهم دوباره به تابع wait; پرش میشود .

بارسیدن cpu به اکولاد ({) برنامه دوباره از اول اجرا میشود و کلیه اعمال بالا از ابتدا انجام میشود .

برای ایجاد تاخیر زمانی شما باید یک تابع ایجاد کنید تا در هنگام تاخیر فراخوانی شود ، برای ایجاد تابع باید ابتدا ان را معرفی کنید :

برای معرفی کلیه توابع از دستور زیر استفاده میشود : (خود برنامه اصلی نیز یک تابع میباشد)

void نام تابع (نوع فضا) ;

در قسمت نام تابع ، شما باید یک نام مناسب بنویسید که در برنامه تابع با همین نام فراخوانی میشود.

در قسمت نوع فضا ، نوع ذخیره سازی تابع مشخص میشود شما میتوانید از عناوین void (فضای خالی) یا جای خالی استفاده کنید .

با این حساب قالب کلی یک برنامه در زبان c به شکل زیر خواهد بود

```
#include <AT۹۱SAMvX۲۵۶.H>                معرفی چیپ
معرفی متغیر ها
void wait(void); معرفی زیر برنامه  (توابعی که قرار است در برنامه فرا  خوانی شوند )
void main(){ تابع اصلی برنامه
ورودی یا خروجی قرار دان پورت ها
while (شروع حلقه){۱)
برنامه اصلی
   wait(); فراخوانی توابع
 } پایان حلقه
} پایان تابع اصلی برنامه
void wait (void) { نام توابعی که قرار است فرا خوانی شود
{ شروع حلقه ی تابع فراخوانی شده
برنامه ای که فراخوانی میشود
} پایان حلقه ی تابع فراخوانی شده (با رسیدن بر نامه به این خط به برنامه اصلی پرش میشود )
```

مقدار زمانی که توسط روش بالا ایجاد یشود از رابطه ی زیر بدست میاید :

زمان =(x*۵)/(f)

X مقدار رقمی است که در حلقه ی for شمرده میشود و f مقدار فرکانس کریستال برحسب هرتز میباشد .

مثال برنامه ای بنویسید که پایه ۱۹.b میکرو کنترلر AT۹۱SAMvX۲۵۶ را به مدت ۲ ثانیه ۰ کند ، بعد به مدت ۱,۵ ثانیه ۱ کند و سپس به مدت ۲,۵ ثانیه ۰ کند و این حلقه مدام تکرار شود .

```
#include <AT۹۱SAMvX۲۵۶.H>
#include <lib_AT۹۱SAMvX۲۵۶.h>
int n;
void wait(void);
int main (void) {
  AT۹۱F_PIO_CfgOutput(AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۱۹);
AT۹۱F_PIO_ClearOutput(AT۹۱C_BASE_PIOB, AT۹۱C_PIO_PB۱۹);
wait();
```

```
wait();
wait();
wait();
AT٩١F_PIO_SetOutput  (AT٩١C_BASE_PIOB, AT٩١C_PIO_PB١٩);
wait();
wait();
wait();
AT٩١F_PIO_ClearOutput(AT٩١C_BASE_PIOB, AT٩١C_PIO_PB١٩);
wait();
wait();
wait();
wait();
wait();
}
void wait (void) {
  for (n = ٠; n < ١٨٤٣٢٠٠; n++);
}
```

برنامه بالا شاید ساده ترین برنامه برای سوال ذکر شده باشد ، در این برنامه ما زمان اصلی ۵۰۰ میلی ثانیه را بوجود اورده ایم و در مکان مناسب به تعداد دفعات مناسب ان را فراخوانی کرده ایم .

تمرین :

۱–به portb.۰ میکرو کنترلر AT٩١SAMvX٢٥٦ یک کلید و به portb.١٩ یک led متصل کرده ایم ، برنامه ای بنویسید که با تحریک کلید led بعد از ۲ ثانیه روشن و با قطع تحریک led بعد از ٤ ثانیه خاموش شود . :

۲–برنامه ای بنویسید ، که پایه های پورت a را یکی از پس از دیگری و با تاخیر ۱ ثانیه روشن کند .

۳–برنامه زیر را تکمیل کنید :

```
#include <AT٩١SAMvX٢٥٦.H>                /* AT٩١SAMTvX٢٥٦ definitions */
#include <lib_AT٩١SAMvX٢٥٦.h>
int n;
int speed;
```

```
void wait(void);
int main (void) {
  AT٩١F_PIO_CfgOutput(AT٩١C_BASE_PIOB, AT٩١C_PIO_PB١٩);
 (AT٩١F_PIO_GetInput(AT٩١C_BASE_PIOB) , AT٩١C_PIO_PB٠) ;
  if (( AT٩١C_PIO_PB٠ == (٠)
  {
  speed++;
 }
AT٩١F_PIO_SetOutput  (AT٩١C_BASE_PIOB, AT٩١C_PIO_PB١٩);
wait();
AT٩١F_PIO_ClearOutput(AT٩١C_BASE_PIOB, AT٩١C_PIO_PB١٩);
wait();
}
void wait (void) {
  for (n = ٠; n < ١٨٤٣٢٠٠; n++);
}
```

این برنامه مربوط به یک چشمک زن میباشد که ما میتوانیم سرعت چشمک زدن led که به پایه ۱۹.b متصل است را توسط کلید های که به پایه های b۰ و b۱ متصل است ، تغییر دهیم .

شما میتوانید جواب مثال های بالا در سایت کویرالکترونیک زیر پیدا کنید :

www.kavirElectronic.ir

چند نکته :

واسط SAM-BA فقط از میکرو کنترلر های زیر پشتیبانی میکند :

- AT٩١SAMv٧S٣٢١
- AT٩١SAMv٧S٦٤
- AT٩١SAMv٧S١٢٨
- AT٩١SAMv٧S٢٥٦

- AT۹۱SAMv۷S۵۱۲
- AT۹۱SAMv۷X۱۲۸ and AT۹۱SAMv۷XC۱۲۸
- AT۹۱SAMv۷X۲۵۶ and AT۹۱SAMv۷XC۲۵۶
- AT۹۱SAMv۷X۵۱۲ and AT۹۱SAMv۷XC۵۱۲
- AT۹۱SAMv۷SE۲۵۶
- AT۹۱SAMv۷SE۵۱۲
- AT۹۱SAMv۷A۳

برای استفاده از واسط SAM-BA برای میکرو کنترلر های AT۹۱SAMv۷S مراحل زیر را انجام دهید :

۱. Power Off the board.
۲. set TST pin to high level.
۳. Power On the board.
٤. Wait about ۱۰ seconds
۵. Power Off the board.
٦. set TST pin to ground.
۷. Power On the board.

نحوه نصب نرم افزار KIEL

این کامپایلر را از یکی از ادرس های زیر تهیه کنید :

دانلود از سرور ۱ :

http://www.ir-manup.com/folder/view/sVUfeSlX

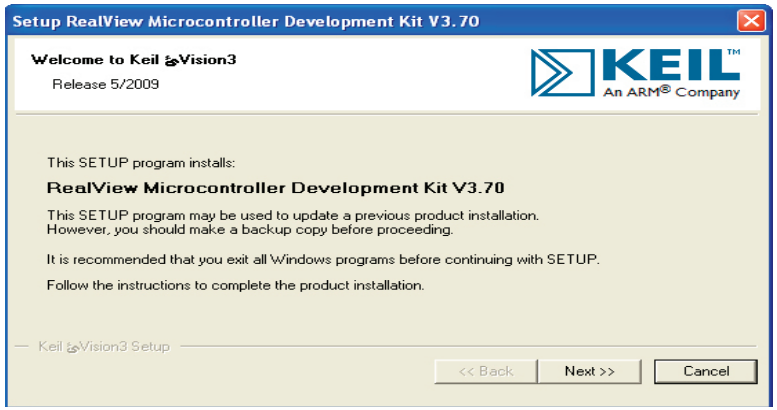
دانلود از سرور ۲ :

http://www.٤shared.com/dir/٩٢/١٨٨٨٣٧٦٥da٧da٦/keil.html

برای تسریع در امر دانلود ما نرم افزار را بر روی دو سرور اپلود کرده ایم .

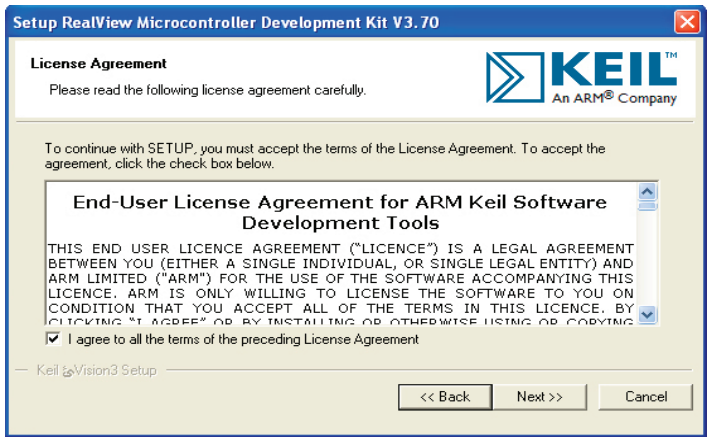
بعد از دانلود کردن نرم افزار ان را UNZIP نمایید برای UNZIP کردن این فایل ها به نرم افزار WINRAR نیاز خواهید داشت شما میتوانید لینک دانلود این نرم افزار را در گوگل بیاید .

بعد از UNZIP کردن نرم افزار بر روی فایل exe.٥_mdk٣٧٠ کلیک کنید تا مراحل نصب اغاز شود :

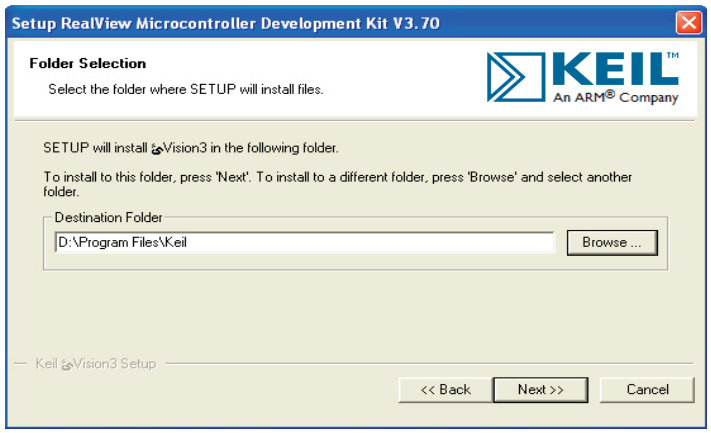


در اولین پنجره بر روی NEXT کلیک نمایید :

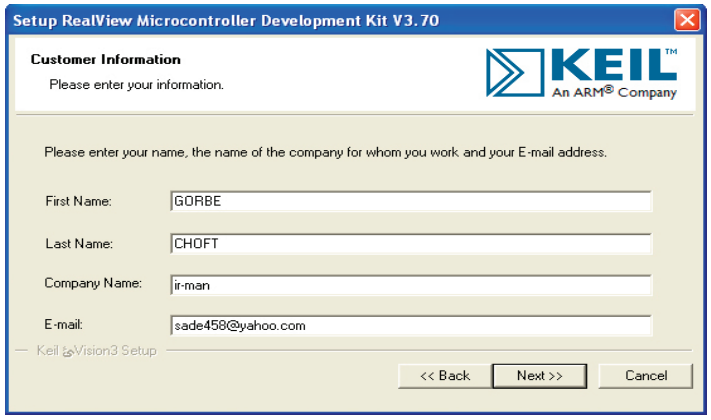
در دومین پنجره با قرارداد مواقفت کرده و بر روی NEXT کلیک کنید :



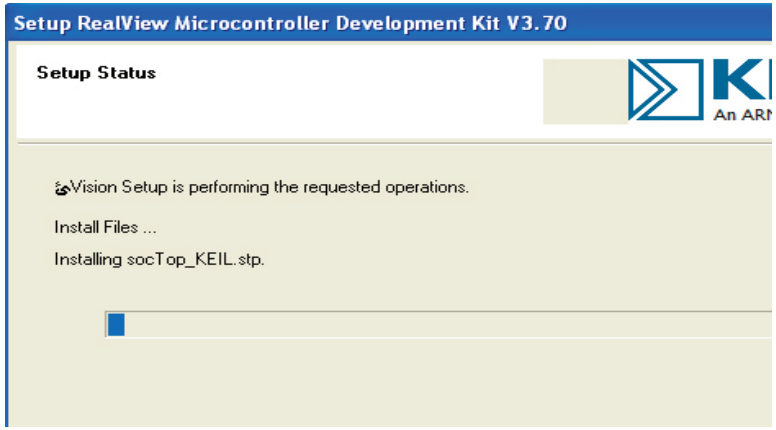
در پنجره سوم مسیر نرم افزار را مشخص کنید و سپس بر روی NEXT کلیک کنید :



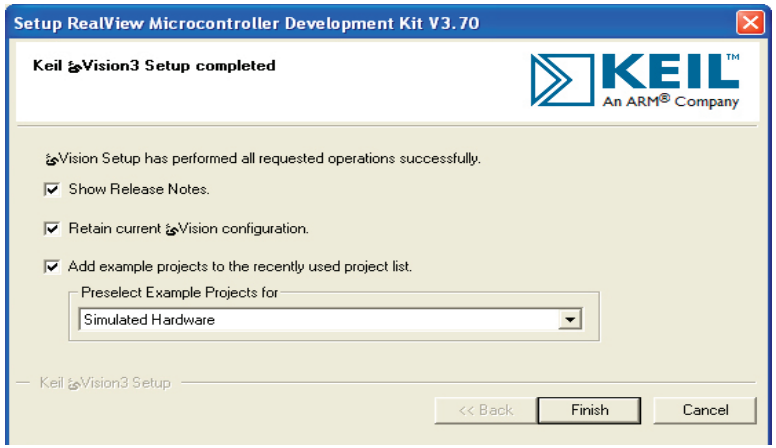
در سومین پنجره اطلاعات مورد نیاز را وارد کنید :



مقداری صبر کنید تا فایل ها کپی شوند :



و در اخرین مرحله بر روی FINISH کلیک نمایید



به پوشه ARM بروید ، در انجا دو پوشه وجود دارد :

محتویات پوشه BIN را در پوشه BIN واقع در محل نصب نرم افزار کپی کنید ، مسیر زیر :

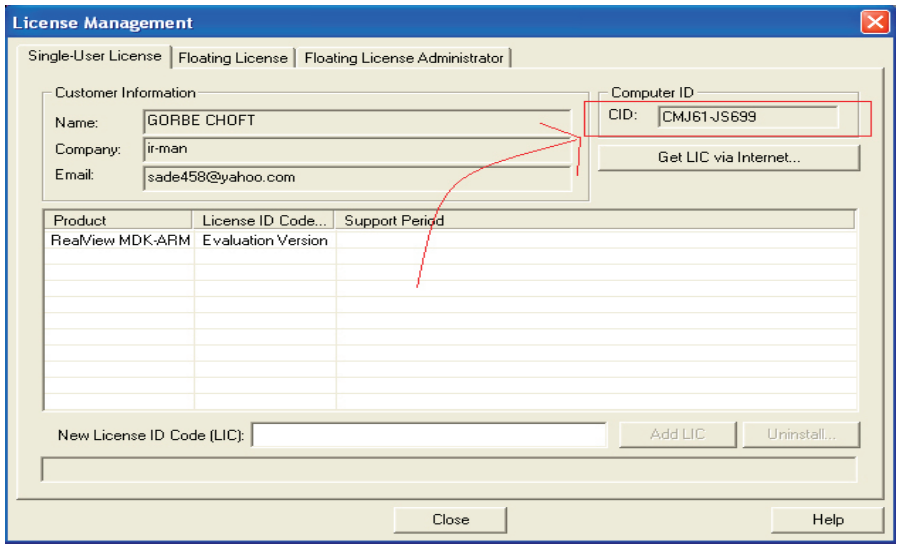
D:\Program Files\Keil\ARM\BIN

محتویات پوشه BIN۳۱ را در پوشه BIN۳۱ یا BIN٤٠ واقع در محل نصب نرم افزار کپی کنید ، مسیر زیر :

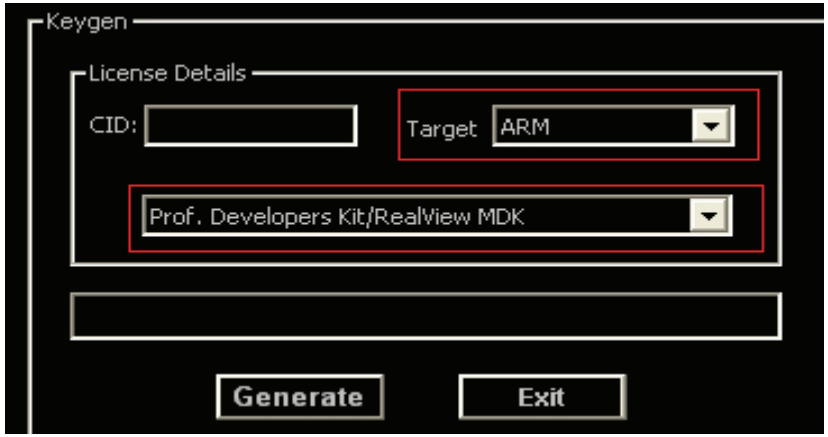
D:\Program Files\Keil\ARM\BIN٤٠

توجه کنید که ادرس بالا ، ادرسی میباشد که من در هنگام نصب برای نرافزار تایین کردم ، این ادرس ممکن است برای شما فرق کند .

نرم افزار را اجرا کنید و از منوی FILE گزینه ی License Management را انتخاب کنید :

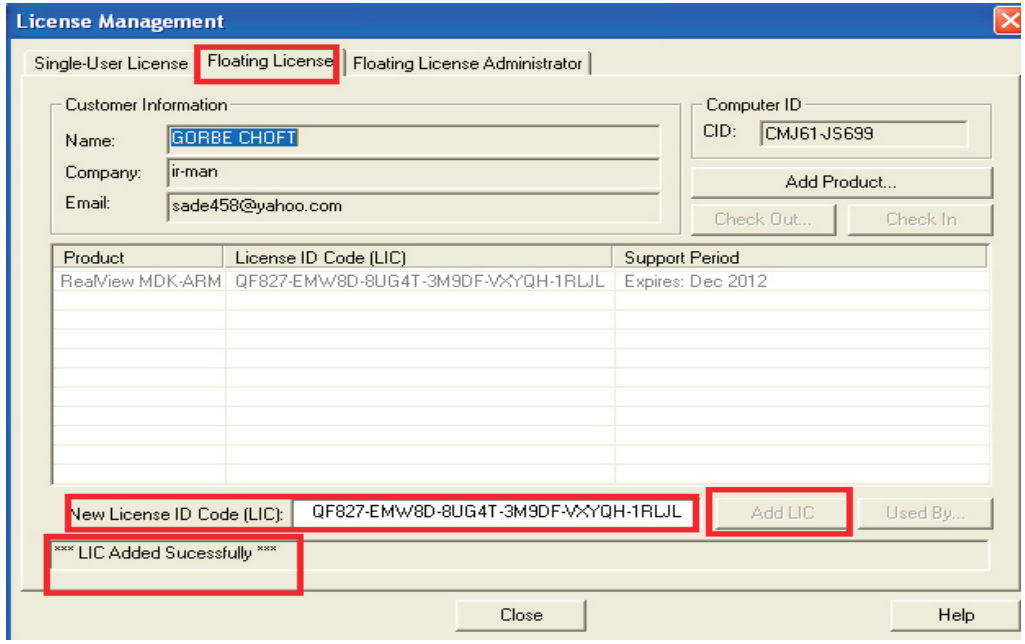


فایل keygen.exe را اجرا کنید و عدد موجود در بخش CID را در ان کپی نمایید و سپس بر روی generate کلیک نمایید ، دقد کنید که تنظیمات keygen مانند تصویر زیر باشد :



شماره سریال تولید شده را کپی کنید و در بخش (new license id code (lic قرار دهید و سپس بر روی گزینه ی add lic کلیک نمایید .
در پنجره License Management تب Floating را انتخاب کنید و شماره سریال را در انجا نیز کپی کنید و بر روی add lic کلیک کنید تا لیسانس پذیرفته شود .

مراحل نصب نرم افزار با موفقیت به پایان رسید . کامپیوتر خود را ریستارت نمایید و با نرم افزار شروع به کار کنید



در صورتی که در هنگام نصب با خطای دمو بود نرم افزار یا ... مواجه شدید ، یکی از مراحل را به درستی انجام نداده اید . مجددا مراحل بالا را تکرار کنید و در صورت تکرار مشکل در انجمن مطرح کنید

اطلاعت AT۹۱SAM۷X۵۱۲ و AT۹۱SAM۷X۲۵۶ و AT۹۱SAM۷X۱۲۸

سه ایسی معرفی شده دقیقا مانند هم میباشند و تنها تفاوت بین انها در مقادیر حافظه میباشد :

ویژگی ها :

Device	Flash	Flash Organization	SRAM
AT91SAM7X512	512 Kbytes	dual plane	128 Kbytes
AT91SAM7X256	256 Kbytes	single plane	64 Kbytes
AT91SAM7X128	128 Kbytes	single plane	32 Kbytes

-کارایی بالا در معماری ۳۲ RISC بیتی

-دستورات ۳۲ بیتی با کارایی بالا

-ICE داخلی (ICE یا In-circuit Emulation به کاربر امکان دیباگ کردن برنامه ، بر روی میکرو را میدهد)

۵۱۲- کیلو بایت حافظه سازمان یافته برای AT۹۱SAM۷X۵۱۲

۲۵۶- کیلو بایت حافظه سازمان یافته برای AT۹۱SAM۷X۲۵۶

۱۲۸- کیلو بایت حافظه سازمان یافته برایAT۹۱SAM۷X۱۲۸

-حافظه فلش این پردازنده ها که مقدار ان در بالا بیان شد ، در دوبانک برای AT۹۱SAM۷X۵۱۲ و یک بانک برای دو میکرو دیگر طراحی شده اند ، هر بانک دارای قطعات ۲۵۶ بایتی میباشد ، این یعنی هر دستور فقط در یک خانه ذخیره میشود و این کار سرعت خواندن حافظه را به مقدار زیادی افزایش میدهد .

-پیشرو در سرعت و اجرای دستورات

۱۲۸-کیلو بایت حافظه sram برای AT۹۱SAM۷X۵۱۲

٦٤- کیلو بایت حافظه sram برای AT۹۱SAM۷X۲۵٦

۳۲- کیلو بایت حافظه sram برایAT۹۱SAM۷X۱۲۸

-سرعت حافظه sram بسیار بالا میباشد ، به طوری که خواندن هر دستور فقط یک سیکل طول میکشد .

-قابلیت بازرسی حافظه (این واحد از نارسایی های که باعث اختلال در عملکرد میشود (معمولا تاخیر در نوشتن و...) ، چشم پوشی میکند)
-واحد کنترل کننده ریست (این واحد در شرایطی که تغذیه اعمال شده مناسب نباشد یا فرمان بازنشانی به پایه ریست اعمال شود ، میکرو را با سرعت بالا ریست میکند)
-درای نوسان ساز rc داخلی با فرکانس ۰ تا ۲۰ مگاهرتز به همراه یک عدد pll
-چندین مد توان (بهینه سازی توان ، کار در فرکانس پایین و مد Idle)
۵- منبع سیگنال پالس خارجی قابل برنامه ریزی
-دو منبع وقفه خارجی (یکی از منابع ، وقفه سریع میباشد و دیگری وقفه عادی)
-دو کانال UART با قابلیت پشتیبانی از دباگ (توسط پروگرامر مخصوص)
-دارای کانتر ۲۰ بیتی قابل برنامه ریزی
-تایمر Watchdog
-درای تایمر بلادرنگ ، این تایمرمیتواند به عنوان یک کانتر ۳۲ بیتی به همراه الارم (متصل به پین مجزا) راه اندازی شود.
-دو پورت ورودی و خروجی موازی
-هر پورت دارای ۳۰ پایه مجزا میباشد .
-امکان استفاده ی هر کدام از پایه های i/o به عنوان ورودی وقفه
-تمامی پایه ها قابلیت برنامه ریزی به عنوان ورودی و با حالت درین باز و مقاومت بالا کشنده را دارند .
-شما میتوانید همزمان یک پایه را به عنوان وردی و خروجی تعریف کنید .
-قابلیت کنترلر ۱۳ وسیله جانبی به صورت مجزا .
-درای یک عدد پورت USB ۲.۲ کامل (با سرعت ۱۲ Mbits بر ثانیه)
-فرستنده گیرنده داخلی قابل برنامه ریزی برای تبادل داده با USB
-دارای یک اترنت MAC ۱۰/۱۰۰ base-T
-امکان رابطه ارتباطی مستقل یا معمولی (MII یا RMII)
-امکان ورود داده و خروج ان به ترتیب ورود (FIFOs) و دسترسی مستقیم به حافظه (DMA) و ارسال و دریافت داده با سرعت بالا
-درای رابط CAN
۸- پیغام قابل برنامه ریزی برای صندوق ایمیل با شمارنده ۱۶ بیتی
-درای یک کنترل کننده سریال هم زمان .
-درای کلاک مستقل و و هم زمان ساز برای هر سیگنال دریافتی و ارسالی
-پشتیبانی از رابط I^2S
-دارای تقسیم گر زمان چند گانه
-سرعت بالا در انتقال داده های ۳۲ بیتی

-پشتیبانی USART از Smart Card ISO۷۸۱۶ T۰/T۱ و Infrared Modulation/Demodulation ® IrDA و Full Modem Line

-درای دو کانال مجزا برای رابط SPI

-یک تایمر / کانتر ۱۶ بیتی ۳ کاناله

-سه کلاک ورودی خارجی

-تولید PWM دوبل و درای مد مقایسه ای و...

-یک واحد کنترل ۱۶ PWM بیتی ۵ کاناله

-پشتیبانی از پروتکل Two-wire

-پشتیبانی از EEPROMs سریال

۸- کانال ۱۰ ADC بیتی ، ۵ کانال Multiplexed

-پشتیبانی از SAM-BA (SAM-BA به کاربر این امکان را میدهد تا از طریق صفحات گرافیکی با میکرو ارتباط برقرار کند)

-پشتیبانی از واسط JTAG

-ولتاژ خروجی هر پورت برابر ولتاژ تغذیه میباشد و هر پین میتواند جریانی تا ۱۶ میلی امپر بدهد .

-عملکرد کاملا ثابت ، کارد کرد در فرکانس ۰ تا ۵۵ مگا هرتز ، و دمای ۴۰- تا ۸۵+ درجه سانتی گراد .

-درای ۱۰۰ پایه و تولید در دو بسته بندی LQFP و TFBGA

وضعیت پایه ها

نوع پایه	عملکردپایه	نام پایه ها در بسته بندی TQFP۱۴۴	شماره
Power	ورودی ولتاژ مرجع ADC	ADVREF	۱
GND	گراند	GND	۲
Analog	ورودی ٤)ADC(	AD٤	۳
Analog	ورودی ٥)ADC(	AD٥	٤
Analog	ورودی ٦)ADC(	AD٦	٥
Analog	ورودی ۷)ADC(	AD۷	٦
نوع پایه	عملکردپایه	نام پایه ها در بسته بندی TQFP۱۴۴	شماره
Power	ولتاژ خروجی رگولاتور (۱.۸ ولت)	VDDOUT	۷
Power	ولتاژ تغذیه	VDDIN	۸
I/O	پورت B.۲۷ / ورودی i.o / (۰) ADC برای تایمر	PB۲۷/AD۰/TIOA۲	۹

I/O	پورت B.۲۸ / ورودی i.o /(۱) ADC برای تایمر	PB۲۸/AD۱/TIOB۲	۱۰
I/O	پورت B.۲۹ / ورودی ۲) ADC)/خروجی پالس	PB۲۹/AD۲/PCK۱	۱۱
I/O	پورت B.۳۰ / ورودی ۳) ADC)/ خروجی پالس	PB۳۰/AD۳/PCK۲	۱۲
I/O	پورت A.۸ / انتخاب مد برنامه ریزی/ درخواست ارسال	PA۸/PGMM۰/RTS۱	۱۳
I/O	پورت A.۹/انتخاب مد برنامه ریزی/ Clear To Send	PA۹/PGMM۱/CTS۱	۱٤
Power ولتاژ تغذیه CPU		VDDCORE	۱۵
GND	گراند	GND	۱٦
Power ولتاژ تغذه خطوط I/O		VDDIO	۱۷
I/O	پورت A.۱۰/ انتخاب مد برنامه ریزی/ خط داده tow-wire	PA۱۰/PGMM۲/TWD	۱۸
I/O	پورت A.۱۱/ انتخاب مد برنامه ریزی/خط کلاک tow-wire	PA۱۱/PGMM۳/TWCK	۱۹
I/O	پورت A.۱۲/خط داده ی برنامه نویسی /cs۰ (پروتکل spi)	PA۱۲/PGMD۰/SPI_NPCS۰	۲۰
I/O	پورت A.۱۳/خط داده ی برنامه نویسی/ cs۱ (پروتکل spi)	PA۱۳/PGMD۱/SPI۰_NPCS۱	۲۱
I/O	پورت A.۱٤/خط داده ی برنامه نویسی/ cs۲ (پروتکل spi)	PA۱٤/PGMD۲/SPI۰_NPCS۲	۲۲
I/O	پورت A.۱۵/خط داده ی برنامه نویسی /cs۳ (پروتکل spi)	PA۱۵/PGMD۳/SPI۰_NPCS۳	۲۳
I/O	پورت A.۱٦/خط داده ی برنامه نویسی/ خط MISO	PA۱٦/PGMD٤/SPI۰_MISO	۲٤
نوع پایه	عملکردپایه	نام پایه ها در بسته بندی TQFP۱٤٤	شماره
I/O	پورت A.۱۷/خط داده ی برنامه نویسی/ خط MOSI	PA۱۷/PGMD۵/SPI۰_MOSI	۲۵
I/O	پورت A.۱۸/خط داده ی برنامه نویسی/ خط کلاک	PA۱۸/PGMD٦/SPI۰_SPCK	۲٦
I/O	پورت B.۹/مدیریت داده ورودی و خروجی (Ethernet)	PB۹/EMDIO	۲۷

I/O	پورت B.۹/مدیریت کلاک (Ethernet)	PB۸/EMDC		۲۸
I/O	پورت B.۹/خط دریافت داده (Ethernet)	PB۱٤/ERX۳		۲۹
I/O	پورت B.۹/ خط دریافت داده (Ethernet)	PB۱۳/ERX۲		۳۰
I/O	پورت B.۹/ خط دریافت داده (Ethernet)	PB٦/ERX۱		۳۱
GND	گراند	GND		۳۲
Power	ولتاژ تغذه خطوط	VDDIO	I/O	۳۳
I/O	پورت B.۵/ خط دریافت داده (Ethernet)	PB۵/ERX۰		۳٤
I/O	پورت B.۱۵/چک اعتبار داده ورودی (Ethernet)	PB۱۵/ERXDV/ECRSDV		۳۵
I/O	پورت B.۱۷/ خط دریافت کلاک(Ethernet)	PB۱۷/ERXCK		۳٦
Power	ولتاژ تغذیه CPU	VDDCORE	۳۷	
I/O	پورت B.۷/ خط دریافت خطا (Ethernet)	PB۷/ERXER		۳۸
I/O	پورت B.۱۲/خط ارسال خطا (Ethernet)	PB۱۲/ETXER		۳۹
I/O	پورت B.۰/خط ارسال کلاک (Ethernet)	PB۰/ETXCK/EREFCK		٤۰
I/O	پورت B.۱/ فعال سازی تبادل داده (Ethernet)	PB۱/ETXEN	٤۱	
I/O	پورت B.۲/ خط ارسال داده (Ethernet)	PB۲/ETX۰		٤۲
نوع پایه	عملکردپایه	نام پایه ها در بسته بندی TQFP۱٤٤		شماره
I/O	پورت B.۳/ خط ارسال داده (Ethernet)	PB۳/ETX۱		٤۳
I/O	پورت B.۱۰/ خط ارسال داده (Ethernet)	PB۱۰/ETX۲		٤٤

I/O	پورت B.۱۱/ خط ارسال داده (Ethernet)	PB۱۱/ETX۳	٤٥
I/O	پورت A.۱۹/خط داده ی برنامه نویسی/ورودی CAN	PA۱۹/PGMD۷/CANRX	٤٦
I/O	پورت A.۲۰/خط داده ی برنامه نویسی/خروجی CAN	PA۲۰/PGMD۸/CANTX	٤٧
Power	ولتاژ تغذه خطوط I/O	VDDIO	٤٨
I/O	پورت A.۲۱/خط ارسال داده (Synchronous)	PA۲۱/PGMD۹/TF	٤٩
I/O	پورت A.۲۲/خط ارسال کلاک (Synchronous)	PA۲۲/PGMD۱۰/TK	٥٠
input	(واسط JTAG)	TDI	٥١
GND	گراند	GND	٥٢
I/O	پورت B.۱٦/اشکارساز اتصال (Ethernet)	PB۱٦/ECOL	٥٣
I/O	پورت B.٤/خط تشخیص حامل(Ethernet)	PB٤/ECRS	٥٤
I/O	پورت A.۲۲//خط ارسال داده (Synchronous)	PA۲۳/PGMD۱۱/TD	٥٥
I/O	پورت A.۲۲// خط دریافت داده (Synchronous)	PA۲٤/PGMD۱۲/RD	٥٦
Input	خط بازنشانی قطعه (ریست)	NRST	٥٧
Input	انتخاب حالت تست	TST	٥٨
I/O	پورت A.۲۲// خط دریافت کلاک (Synchronous)	PA۲٥/PGMD۱۳/RK	٥٩
I/O	پورت A.۲۲// خط دریافت داده (Synchronous)	PA۲٦/PGMD۱٤/RF	٦٠
نوع پایه	عملکردپایه	نام پایه ها در بسته بندی TQFP۱٤٤	شماره
Power	ولتاژ تغذه خطوط I/O	VDDIO	٦١
Power	ولتاژ تغذیه CPU	VDDCORE	٦٢

I/O	پورت B.۱۸/ Force ۱۰۰ Mbits.sec	PB۱۸/EF۱۰۰	٦٣
I/O	پورت B.۱۹/ PWM۰	PB۱۹/PWM۰	٦٤
I/O	پورت B.۲۰/ PWM۱	PB۲۰/PWM۱	٦٥
I/O	پورت B.۲۱/ PWM۲	PB۲۱/PWM۲	٦٦
I/O	پورت B.۲۲/ PWM۳	PB۲۲/PWM۳	٦٧
GND	گراند	GND	٦٨
I/O	پورت B.۲۳/ i.o برای تایمر	PB۲۳/TIOA۰	٦٩
I/O	پورت B.۲٤/ i.o برای تایمر	PB۲٤/TIOB۰	۷۰
I/O	پورت B.۲٥/ i.o برای تایمر	PB۲٥/TIOA۱	۷۱
I/O	پورت B.۲٦/ i.o برای تایمر	PB۲٦/TIOB۱	۷۲
I/O	پورت A.۲۲// خط ارسال داده Debug	PA۲۷/PGMD۱٥/DRXD	۷۳
I/O	پورت A.۲۸/ خط دریافت داده Debug	PA۲۸/DT X D	۷٤
I/O	پورت A.۲۹/ ورودی وقفه فوری	PA۲۹/FIQ	۷٥
Output	Test Data Out(واسط JTAG)	TDO	۷٦
Input	انتخاب گر JTAG	JTAGSEL	۷۷
Input	Test Mode Select (واسط JTAG)	TMS	۷۸
نوع پایه	عملکردپایه	نام پایه ها در بسته بندی TQFP۱٤٤	شماره
Input	Test Clock(واسط JTAG)	TCK	۷۹
I/O	پورت A.۳۰/ورودی وقفه خارجی	PA۳۰/IRQ۰	۸۰

I/O	پورت A.۰ / / ورودی داده سریال USART۰	PA۰/PGMEN۰/RXD۰		۸۱
I/O	پورت A.۱ // خروجی داده سریال USART۰	PA۱/PGMEN۱/TXD۰		۸۲
GND	گراند	GND		۸۳
Power	ولتاژ تغذه خطوط I/O	VDDIO	۸٤	
I/O	پورت A.۳/ درخواست ارسال	PA۳/RTS۰		۸۵
I/O	پورت A.۲/ خط کلاک سریال	PA۲/SCK۰		۸٦
Power	ولتاژ تغذیه CPU	VDDCORE		۸۷
I/O	پورت A.٤// Clear To Send	PA٤/PGMNCMD/CTS۰		۸۸
I/O	پورت A.۵ // ورودی داده سریال USART۱	PA۵/PGMRDY/RX D ۱		۸۹
I/O	پورت A.٦// خروجی داده سریال USART۱	PA٦/PGMNOE/T X D۱		۹۰
I/O	پورت A.۷ / مسیر هدایت داده / خط کلاک سریال	PA۷/PGMNVD/SCK۱		۹۱
Input	پاک کردن فلش	ERASE		۹۲
Analog	+USB	DDM		۹۳
Analog	-USB	DDP		۹٤
Power	ولتاژ تغذیه فلش و USB	VDDFLASH		۹۵
GND	گراند	GND		۹٦
نوع پایه	عملکردپایه	نام پایه ها در بسته بندی TQFP۱٤٤		شماره
Input	ورودی کلاک به میکرو از کریستال	XIN/PGMCK		۹۷
Output	خروجی کلاک ازمیکرو به کریستال	XOUT		۹۸

Input	اتصال به فیلتر RC مربوط به PLL	PLLRC	۹۹
Power	ولتاژ تغذیه PLL	VDDPLL	۱۰۰

نکات :

ولتاژ تغذیه ۱.۸ cpu ولت میباشد ، شما میتوانید این ولتاژ را از پایه VDDOUT تامین کنید ، این ولتاژ باید به پایه VDDCORE اعمال شود .

ولتاژ تغذیه سایر بخش های میکرو ۳.۳ ولت میباشد ، این ولتاژ باید از طریق رولاتور های تثبیت ولتاژ تامین شود و به پایه های VDDPLL و VDDIO و VD-FLASH اعمال میشود .

شما میتوانید به پایه های وردی ولتاژی بین ۲ تا ۵.۵ ولت به عنوان سطح منطقی ۱ و ولتاژ ۰ تا ۱.۵ ولت را به عنوان سطح منطقی صفر اعمال کنید .

ولتاژ خروجی پایه ها ۳.۳ ولت با حداکثر جریان ۱۶ میلی امپر است .

برد آموزشی میکروکنترلر ARM سری AT91SAM7X
راهنمای کاربران

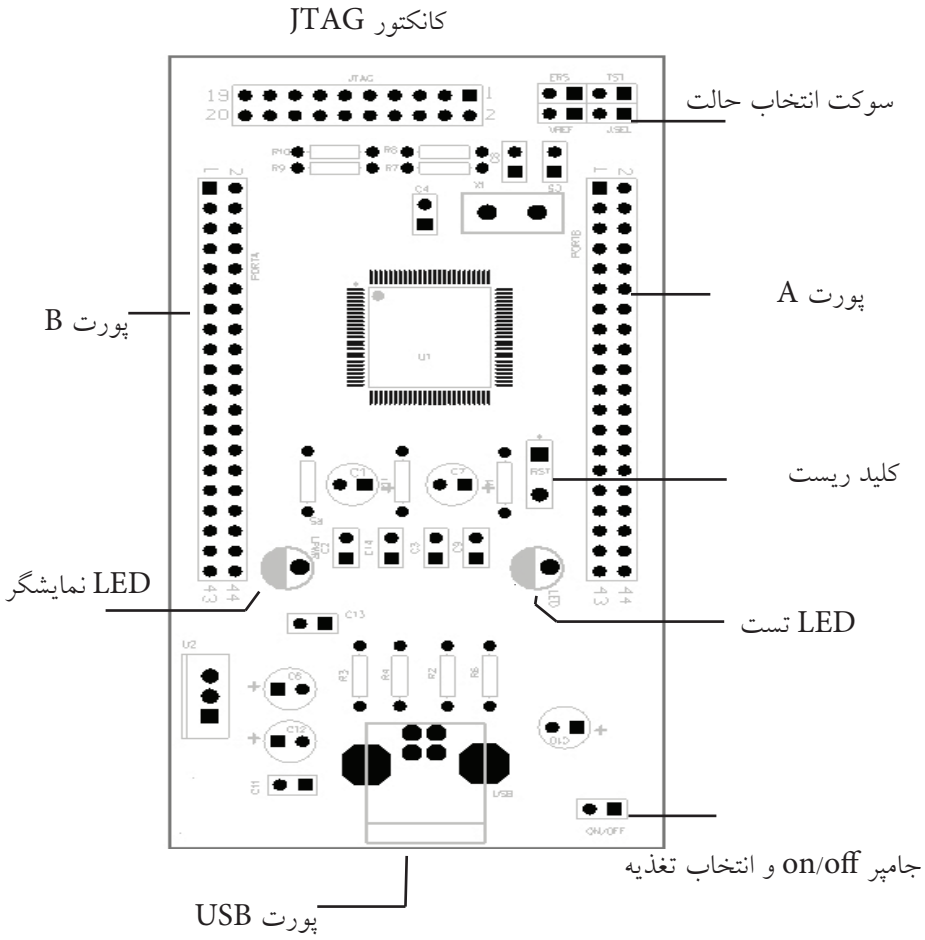
این برد در جهت تسهیل در آموزش ها طراحی شده است و مبنای آموزش های ما همین برد خواهد بود همچنین این برد با قیمت بسیار مناسب برای کاربران با تخفیف های ویژه ارائه شده است

این محصول به صورت زیر ارسال می گردد:
۱.برد مونتاژ شده به صورت کامل (میکرو روی برد AT۹۱SAM۷X۲۵۶)
۲.کابل USB
۳.دو عدد IDC ۴۰ پین
۴.یک عدد IDC ۲۰ پین
۵.کابل فلت جهت مونتاژ IDC ها
۶. dvd نرم افزاری شامل ، اخرین ورژن نرم افزار keil ، دیباگر hjtag ، نرم افزار ISP v۱,۱۳-AT۹۱ ، دیتاشیت ها و برگه های اطلاعاتی شرکت اتمل و.... (۱dvd)

نکته ها :

کلیه قطعات مورد نیاز برد بر روی انها نصب شده است . (تمامی مواردی که در بخش های قبل تحت عنوان امکانات برد اموزشی به انها اشاره شد).
برد مونتاژ نشده نیز موجود می باشد
گروه ما اموزش ARM را تا انتها ادامه خواهد داد ، ما نحوه کار با کلیه امکانات برد را اموزش خواهیم داد .
به زودی پروگرامر jtag و سایر لوازم مربوطه نیز به فروشگاه اضافه میگردد .
در صورتی که در هنگام کار با برد با مشکلی روبرو شدید به ادرس های پشتیبانی که در انتها اورده شده است مراجعه کنید .

تصویر برد از روبرو:



پورت B و پورت A :

کلیه پایه های پورت a و پورت b از طریق سوکت های مشخص شده ، در دسترس شما قرار دارد.

شما میتوانید این پایه ها را از طریق کابل های IDC به برد اصلی منتقل کنید یا انها را بر روی برد برد بیاورید و با انها کار کنید .

LED نمایشگر(قرمز) :

روشن بودن این led نمایانگر درست بودن ولتاژ تغذیه و روشن بودن برد میباشد . در صورتی که این led خاموش باشد ، در تغذیه برد مشکلی بوجود امده است .

LED تست(زرد) :

این led به پایه ۱۹ از پورت b متصل شده است ، وظیفه این led تست کردن برد میکرو و امکانات ان میباشد . قبلا در مورد نحوه تست پایه ها و برنامه ریزی میکرو بحث کردیم .

کانکتور JTAG :

برای اتصال واسط jtag به برد از این کانکتور استفاده میشود .در این کانکتور ولتاژ ۳,۳ ولت برای تغذیه مدار پروگرامر تعبیه شده است ، ولتاژ از طریق کابل jtag به پروگرامر منتقل میشود .

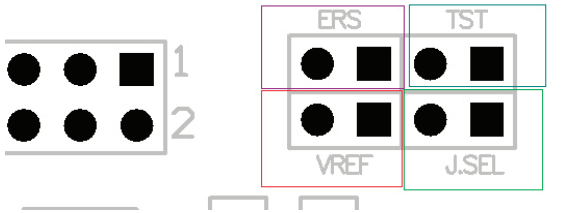
کلید ریست :

برای بازنشانی دستی میکرو از این کلید استفاده میشود . با فشردن این کلید میکرو ریست میشود.

جامپر انتخاب برد (روشن/خاموش) :

این جامپر نقش کلید on/off را در صورت استفاده از تغذیه usb بر عهده دارد در صورتی که مایلید میکرو از طریق برد اصلی تغذیه شود این جامپر را بردارید ولتاژ ۳,۳ ولت از طریق کابل های IDC به برد منتقل میشود.

بخش «سوکت انتخاب حالت»



در این بخش تعداد ٤ عدد پین هدر دوبل وجود دارد :

ERS : در صورتی که جامپر را بر روی این پین هدر قرار دهید ، حافظه فلش میکرو پاک خواهد شد .

برای پاک کردن حافظه فلش ، ابتدا تغذیه برد را متصل کنید ، در مرحله دوم جامپر را بر روی این پین هدر قرار دهید ، در مرحله سوم ، تغذیه را قطع کنید و در نهایت جامپر را بردارید ، حافظه میکرو پاک شد .

tst : از این جامپر برای اعمال مد برنامه ریزی موازی و تست میکرو در هنگام ساخت استفاده میشود .

jsel : این جامپر توانا ساز پورت jtag میباشد . در صورتی که قصد استفاده از پروگرامر jtag را دارید این جامپر را متصل کنید .

vref : این جامپر مربطو به تنظیم کردن ولتاژ مرجع adc میباشد . با متصل کردن این جامپر ولتاژ مرجع بر روی vcc تنظیم میشود (در اینده در این مورد بیشتر بحث خواهیم نمود .

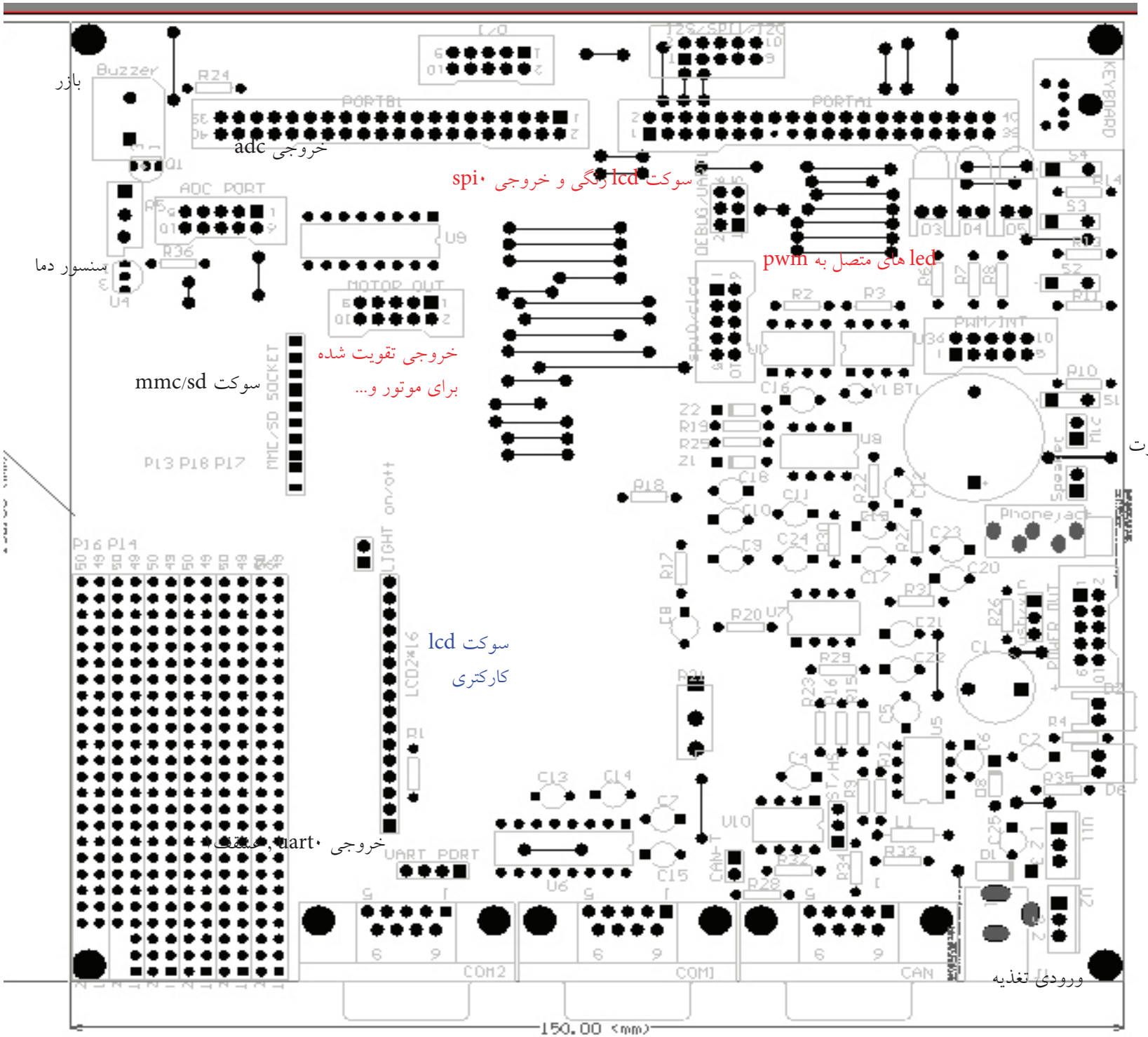
تصویر برد اصلی در صفحه بعد اورده شده است .

در مباحث قبلی به بررسی ویژگی های برد پرداختیم . توضیح در مورد ویژگی ها را به شماره های بعد و در هنگام اموزش کار با پروتکل موکول میکنیم .

برای کار با این برد نیاز به انجام تنظیمات خاصی نیست ، کافی است کابل های IDC را به برد میکرو و برد اصلی متصل کنید .(به دلیل استفاده از سوکت های مخصوص ، امکان برعکس زدن پورت ها وجود ندارد)

ولتاژ تغذیه برد اصلی بین ۷,۵ تا ۱۲ ولت مباشد . شما همچنین میتوانید ولتاژ تغذیه را از طریق پروت usb تامین کنید .

کلیه امکانات نظیر lcd رنگی ، کاراکتری و... بر روی برد نصب می شود .


خروجی ۸ بیتی
خروجی spi۱ و i۲c و i۲s
محل اتصال کیبرد
بازر
کلید های فشاری
خروجی adc
سوکت lcd وقفگی و خروجی spi۰
led های متصل به pwm
سنسور دما
خروجی pwm و
ورودی وقفه
خروجی تقویت شده
برای موتور و...
سوکت mmc/sd
ورودی و خروجی صوت
خروجی تغذیه برای
دیگر لوازم جانبی
سوکت lcd
کارکتری
led های نمایشگر
خروجی uart۰ و وقفه
ورودی تغذیه
پورت های com۱ و com۲
پورت can

برای راحتی دوستانی که در شهرستان هستند بر آن شدیم چندین روش برای فروش محیا کنیم
ارسال برد به تمام ایران امکان پذیر می باشد و هیچ گونه محدودیتی برای دوستانی که در شهرستان هستند نداریم

روش های تهیه برد اموزشی

خرید نقدی و تحویل به ترمینال (پیشنهاد ما–سریعترین روش و بهترین روش می باشد)
پرداخت وجه از طریق بانک (کارت به کارت– واریز مستقیم به حساب از طریق بانک)
در این روش سفارش شما به ترمینال تحویل داده می شود و حداکثر ۲۴ ساعته بدست مشتری می رسد
شامل تخفیف می شود

خرید نقدی و ارسال با پست پیشتاز (پیشنهاد ما)
پرداخت وجه از طریق بانک (کارت به کارت– واریز مستقیم به حساب از طریق بانک)
زمان تحویل ۳–۴ روز کاری
شامل تخفیف می شود

خرید مستقیم:
یزد – میبد خیابان شهید میرباقری روبروی دبیرستان کارودانش آینده سازان –کویرالکترونیک
شامل تخفیف می شود
در صورتیکه تمایل به دریافت نمایندگی محصولات ما دارید با شماره های تماس هایی که در آخرین صفحه هست تماس بگیرید
خرید پستی (پرداخت هزینه به مامور پست)
این سیستم به دو صورت ارسال می گردد به صورت سفارشی و پیشتاز که پس از دریافت محصول هزینه به مامور پست مطابق فاکتور تحویل می گردد

شامل تخفیف نمی شود

جهت سفارش برد و راهنمایی های بیشتر به فروشگاه اینترنتی کویرالکترونیک مراجعه فرمائید
به زودی برد اصلی هم تولید و در فروشگاه قرار خواهد گرفت

www.kavirelectronic.ir/shop

این اموزش توسط گروه کویر الکترونیک در شرایطی که منبع فارسی و انگلیسی مناسبی برای ARM وجود ندارد تهیه شده است و ممکن است به مرور زمان دچار تغییر و تحول شود . در صورت کپی برداری از مطالب حتما منبع را ذکر کنید تا سایر خوانندگان گمراه نشوند .

WWW.KAVIRELECTRONIC.IR/FA

Support :09138543207-0352-7730313

Email: info@kavirelectronic.ir

Email: kavirelectronic.com@gmail.com

Yahoo ID : www.kavirelectronic@yahoo.com

جهت دریافت آخرین اخبار این ای دی یاهو را به لیست خود اضافه کنید