اموزش میکرو کنترولر های XMEGA

با کامپایلر CODEVISION AVR

مقدمه:

با رشد وگسترش میکروکنترولر ها در دنیای امروز و نیاز مبرم به این قطعه و استفاده ان در صنعت ،یادگیری و استفاده از ان نیز پیش از پیش لازم و در اولویت به نظر می رسد.به همین دلیل بنده تصمیم گرفتم اموزشی در این باب برای میکروکنترو لرهای XMEGA تهیه کنم.در این اموزش هدف این می باشد که با راحترین و بهترین شیوه مطلب مورد نظر ارائه شود به طوری که قابلیت استفاده از ان برای اکثریت وجود داشته باشد.کامپایلر مورد استفاده در این اموزش CODEVISION AVR میباشد که یک کامپایلر متوسط به حساب می اید که خود بنده از IAR استفاده میکنم ،البته از تمام کسانی که قصد اموزش اصولی و پایه این میکرو را دارند توصیه میکنم از کامپایلر هائی مانند WINAVR,IAR استفاده نمایند.چون در این اموزش از بیان مطالب اضافی جلوگیری خواهد شد لذا از تمام خوانندگان انتظار می رود با برنامه نویسی به زبان C و کامپایلر CODEVISION اشنائی نسبی داشته باشند.با امید اینکه اموزش مورد نظر مورد توجه خوانندگان عزیز قرار گیرد.

نویسنده : پوریا علیزاده

XMEGA:

XMEGA سری جدید خانواده AVR است که توسط کمپانی ATMEL عرضه شده است و در عین سازگاری کامل از نظر کدنویسی، دارای توانایی و امکانات بسیار بیشتری نسبت به گروه های S90، Tiny و Mega می باشد. این سری از محصولات جدید با دارا بودن امکاناتی بسیار قوی حتی رقیب قدرتمندی برای میکروکنترلرهای ARM7 محسوب می شود و به همین دلیل قابلیت کاربرد در تولیدات مختلف صنعتی را در سطوح مختلف دارا می باشد.

وجه مشخصه اصلی این خانواده در چند مورد خلاصه می شود:

1- سرعت بالاتر در انجام عملیات که در درجه اول ناشی از حداکثر کلاک قابل اعمال به CPU و سخت افزارهای جانبی است. فرکانس کلاک در این خانواده حداقل MHz32 است که در عمل با Overclock به مقادیر بیشتری هم می توان رسید. عامل دوم وجود امکاناتی مانند Event system و DMA است که راندمان نرم افزار را در یک کلاک برابر به میزان قابل توجهی افزایش می دهند و سبب کاهش بار CPU برای انجام بسیاری از عملیات می شوند.

2- سخت افزارهای جانبی بسیار غنی مانند 8 عدد USART و 4 عدد TWI و 4 عدد SPI و ADC و DAC با دقت 12 بیت و واحد های رمزنگاری AES و DES و مقایسه کننده های آنالوگ و امکان اتصال به SDRAM خارجی و 8 عدد تایمر 16 بیتی با 24 خروجی PWM و موارد متعدد دیگری که برای این خانواده اهمیت خاصی را ایجاد کرده است.

3- مصرف توان بسیار پائین که استفاده از XMEGA را در کاربردهایی که مصرف توان در آن مهم است، کاملا توجیه پذیر می کند. در یکی از شماره های این خانواده امکان اتصال یک Battery backup خارجی وجود دارد و با قطع تغذیه واحد RTC32 داخلی همچنان به عملیات زمانگیری خود ادامه می دهد.

بنابراین با توجه به این مزیت ها و شباهت هایی که در عملکرد XMEGA با خانواده های قبلی وجود دارد، بسط و گسترش اطلاعات این میکروکنترلر جدید از اهمیت خاصی برخوردار می باشد.

-XMEGA آشنایی با سری A وD

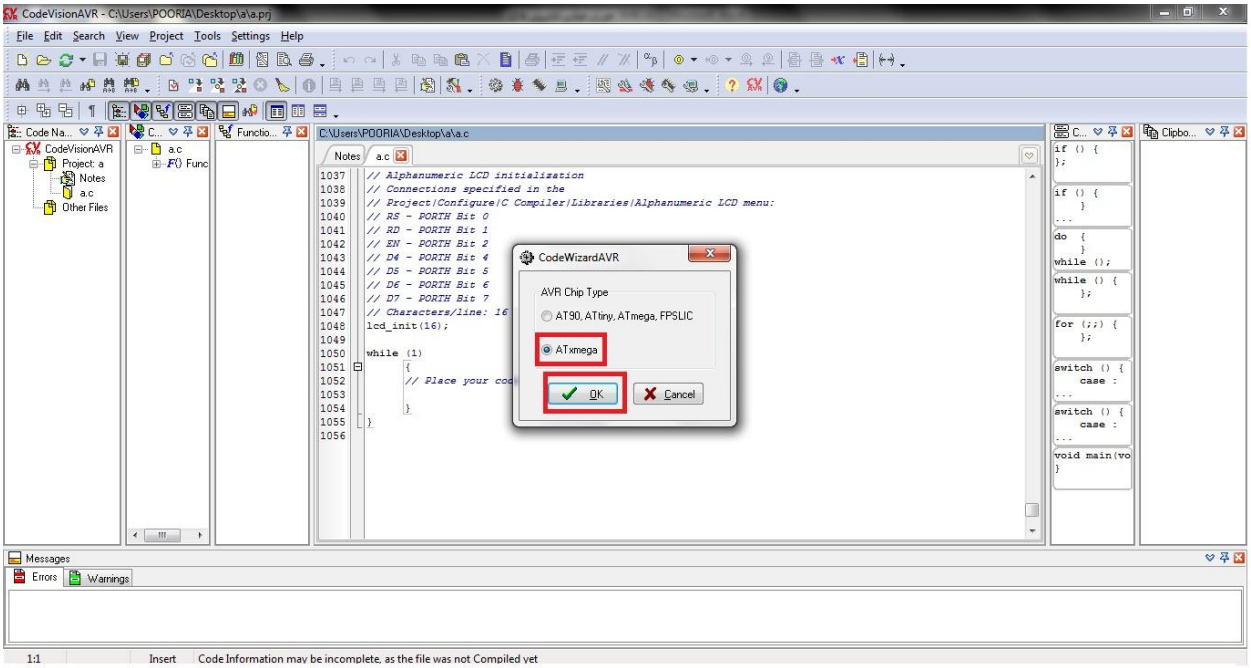
خانواده XMEGA تا این زمان در دو سری کلی A و D عرضه شده اند. سه زیر گروه A1 و A3 و A4 برای سری A و دو زیرگروه D3 و D4 برای سری D وجود دارند. به طور کلی امکانات سری A از سری D قوی تر است و در هر گروه هم شماره های با عدد انتهای کوچکتر دارای امکانات بیشتری هستند. در متن شماره هر IC مانند سری های Mega مقدار Flash آن شماره ذکر می شود. با این توضیحات ATXMEGA384A1 به عنوان قوی ترین شماره این خانواده و ATXMEGA16D4 به عنوان ضعیف ترین شماره این خانواده شناخته می شوند. از نظر Package این شماره ها در 3 نسخه 44 و 64 و 100 پایه عرضه می شوند. سری های D4 و A4 دارای 44 پایه و D3و A3 دارای 64 پایه و سری A1 دارای 100 پایه هستند. برای این خانواده نسخه DIP وجود ندارد و همگی بصورت SMD هستند. برای انتخاب هر شماره باید با توجه به امکانات داخلی IC و نیازهای طراحی و موجود بودن در بازار اقدام شود و قبل از انتخاب یک شماره نسبت به واحدهای سخت افزاری موجود در آن بررسی کاملی به عمل آید.

به عنوان مثال در شماره های سری D اصولا DMA و DAC وجود ندارد و عدم توجه به این مسئله می تواند انجام یک منظور از پیش تعیین شده را که مستلزم استفاده از این امکانات است با مشکل مواجه کند.

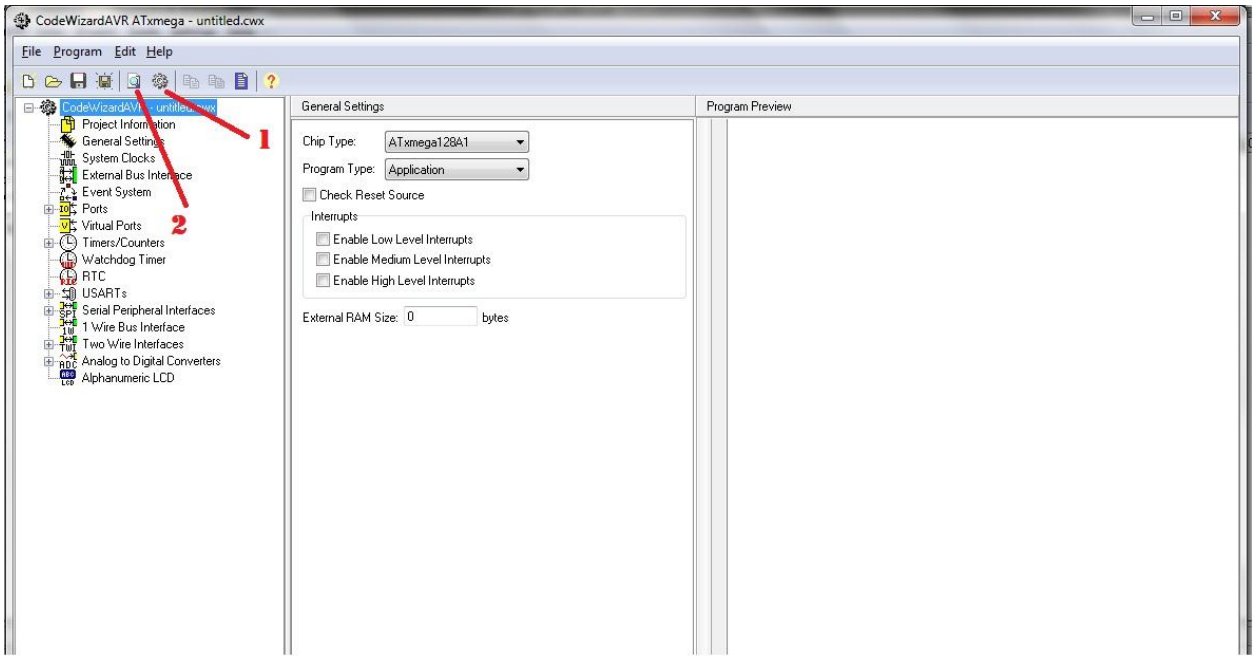
البته در زمان نگارش این اموزش سری جدیدی از این میکرو با پسوند AU ارائه شده اند که قابلیت اتصال به پورت USB با سرعت بالا را دارند.

ایجاد اولین پروژه با CODEVISION:

بعد از باز کردن نرم افزار به سربرگ file رفته و گزینه new را انتخاب کنید،در کادر باز شده گزینه project را انتخاب کرده و تائید نمائید ودوباره در کادر ظاهر شده پیغام مورد نظر را تائید نمائید بعد از انجام این مراحل کادر دیگری باز می شود که شما باید در این قسمت سری میکرو کنترولر خود را انتخاب کنیدکه ما از این قسمت xmega را انتخاب می نمائیم و تائید میکنیم.



بعد از این مرحله قسمتی با نام codewizard باز می شود که در این قسمت می توانید تنظیمات اولیه میکرو مانند تنظیم کلاک ،پورت هاوتایمر هاو.... را انجام دهید:

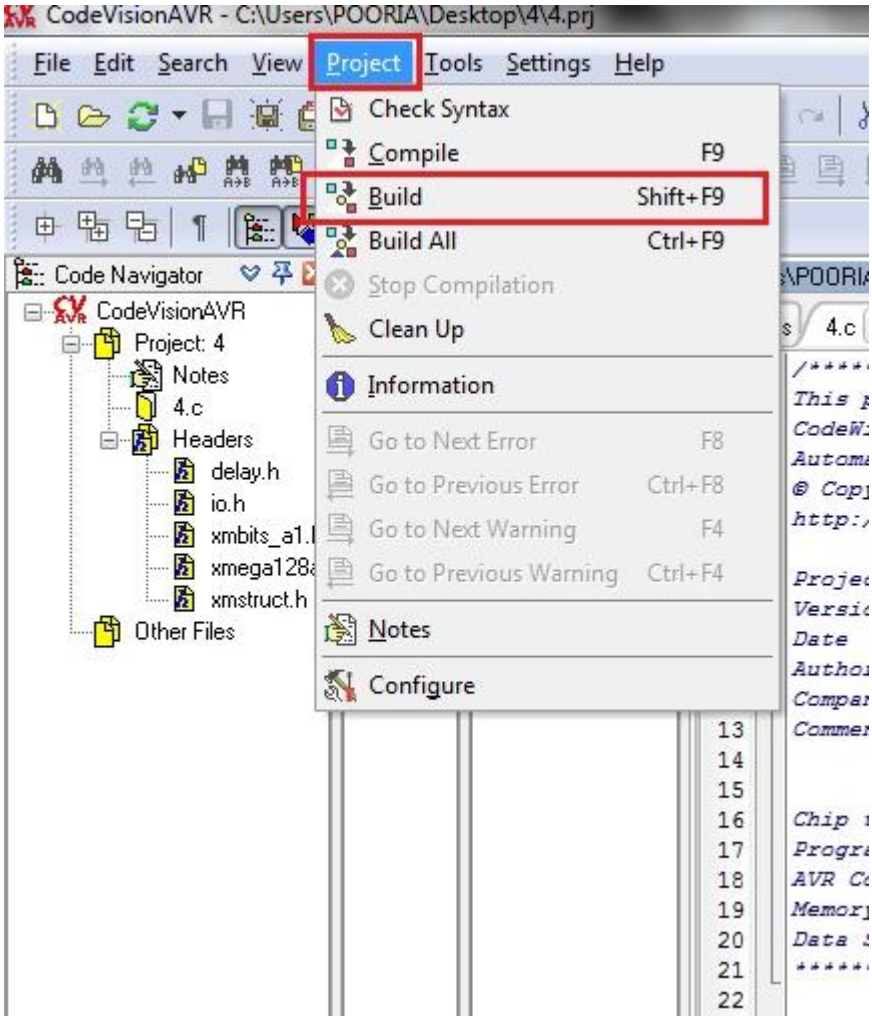


1-بعد از اینکه تنظیمات مورد نظر را انجام دادید با انتخاب این گزینه کامپایلر از شما 3 بار برای ذخیره پروژه اسم و مکان ذخیره پروژه و فایلهای مورد نیاز را می خواهد که باید هر 3 بار با یک نام و در یک جا ذخیره شوند.

2-با انتخاب این گزینه قبل از اینکه فایل خود را ذخیره کنید می توانید تنظیمات انجام شده توسط codewizard را مشاهده کنید.

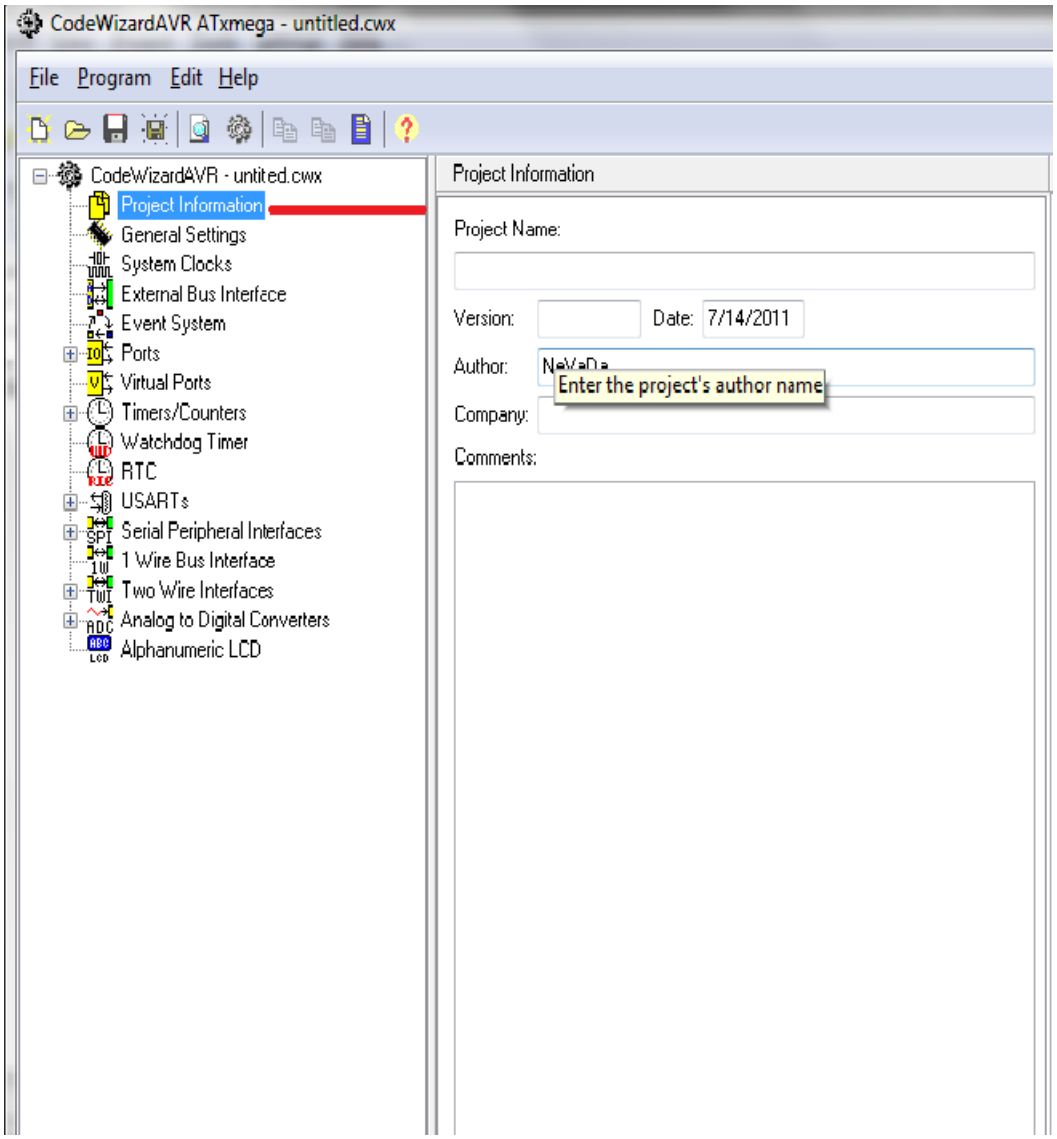
کامپایل کردن برنامه:

بعداز اینکه تنظیمات مورد نظر را انجام دادید و سورس مورد نظر خود را نوشتید با زدن گزینه BUILD در منوی PROJECT پروژه شما کامپایل شده و فایل هگز مورد نظر ساخته می شود.



منوی project information در codewizard:

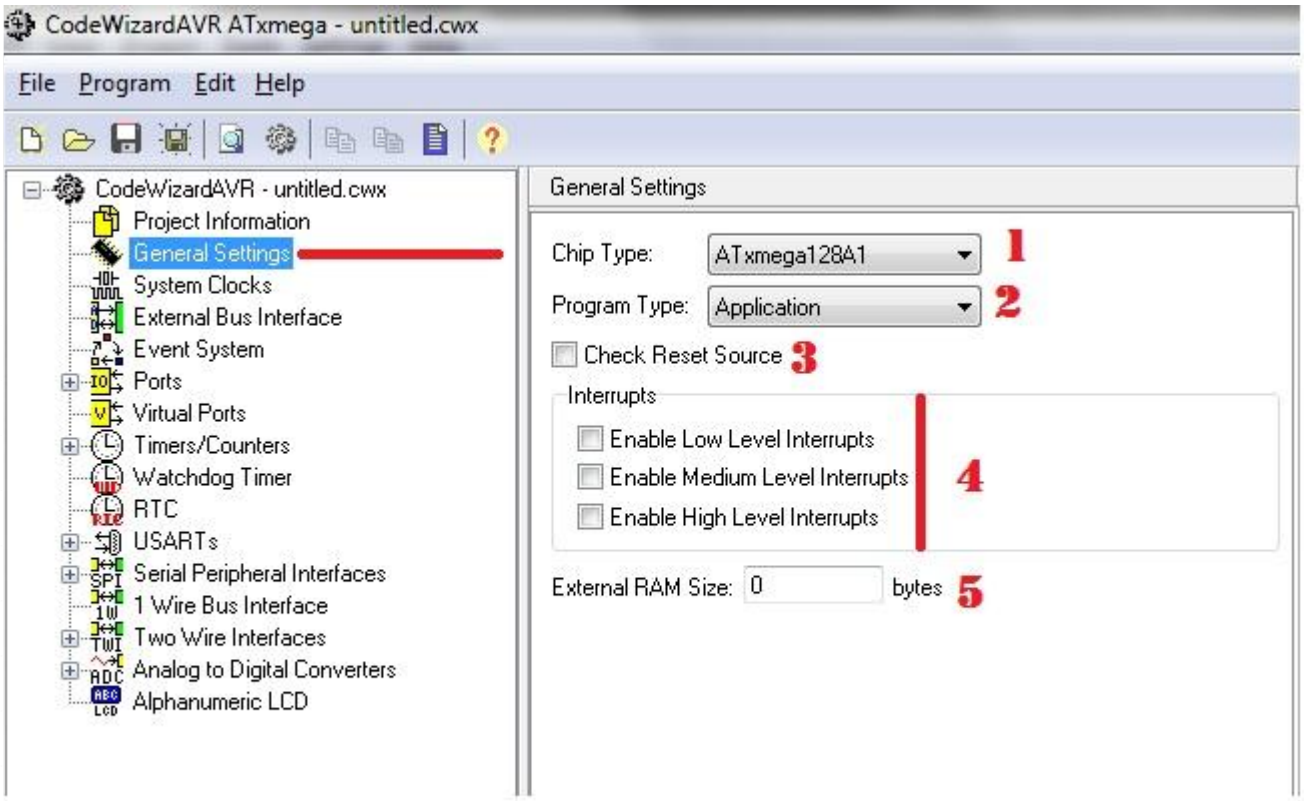
در این قسمت می توانید توظیهات پروژه مانند تاریخ ایجاد ،نام پروژه و... را اضافه نمائید.



منوی general setting:

شما در این قسمت می توانید تنظیمات عمومی مانند انتخاب تراشه،حافظه مورد استفاده،منابع ریست و منابع وقفه را انجام دهید.

-



1-در این قسمت نوع میکروی مورد استفاده مشخص می گردد.(که مبنای اموزش ما ATXMEGA128A1 می باشد)

2-در این قسمت حافظه مورد استفاده تعیین می گردد(که به صورت پیش فرض درست است)

3-در این قسمت می توانیم منابع ریست میکرو را فعال یا غیر فعال کنیم.(بصورت پیش فرض غیر فعال است)

4-در این قسمت می توانیم منابع وقفه را که به 3 سطح تقسیم می شوند فعال کنیم.(بصورت پیش فرض غیر فعال هستند،توظیهات بیشتر در قسمت منابع وقفه)

5-در این قسمت اگر از RAM خارجی استفاده میکنیم سایز انرا بر حسب BAYTE مشخص می کنیم.

XMEGA – منابع Clock

منابع تامین پالس ساعت برای CPU و سایر سخت افزارها در این خانواده متعدد هستند. 5 منبع شامل اسیلاتور 2 مگاهرتز داخلی، اسیلاتور 32 مگاهرتز داخلی، اسیلاتور 32.768 کیلوهرتز داخلی، خروجی PLL و اسیلاتورها یا کریستال های خارجی به عنوان منابع تامین کلاک قابل انتخاب هستند. حداکثر کلاک نامی برای این خانواده از نظر اعمال به CPU در حد 32 مگاهرتز است که تا سرعت اجرای حداکثر 32 ملیون دستورالعمل در ثانیه را تامین می کند. اما در عمل با افزایش کلاک می توان به مقادیر بیشتر هم رسید که به این عمل Over Clock گفته می شود. بر خلاف اکثر شماره های AVR که تغییر منبع کلاک نیازمند برنامه ریزی فیوزبیتهاست، این امر در XMEGA توسط خطوط برنامه انجام می شود و در هر زمان که لازم باشد می توان منبع کلاک را تغییر داد. برای واحد های سخت افزاری داخلی می توان کلاکی متفاوت از

کلاک CPU تعریف کرد و این امر از طریق تقسیم کننده های قابل برنامه ریزی که به همین منظور پیش بینی شده اند، میسر است. دو واحد سخت افزاری در XMEGA وجود دارند که یکی تا 64 مگاهرتز و دیگری تا 128 مگاهرتز کلاک را می پذیرند. برای استفاده از این ظرفیت باید ابتدا با PLLداخلی یک فرکانس 128 مگاهرتز تولید کرد و با تنظیمات لازم دوبار آن را بر دو تقسیم کرد. حال این 3 فرکانس متمایز یعنی 128 و 64 و 32 مگاهرتز هر یک به قسمت مربوط به خود اعمال می شود که 32 مگاهرتز مربوط به CPU است.

باید به این نکته توجه شود که بالاتر بودن کلاک به خودی خود ملاک کاملی برای سرعت XMEGA نیست و وجود واحدهایی مانند Event System و DMA می تواند سرعت اجرای عملیات را به ازای یک کلاک ثابت بسیار بالاتر ببرد و این مسئله ای است که باید در کنار فرکانس کلاک بصورت توام به آن توجه شود.

اسیلاتور داخلی 2 مگاهرتز: بعد از Reset شدن میکروکنترلر، این اسیلاتور بصورت خودکار به عنوان منبع کلاک سیستم انتخاب می شود. همچنین مضاربی از فرکانس این اسیلاتور از طریق PLL می تواند به عنوان کلاک انتخاب شود.دقت این اسیلاتور در دمای 25 درجه سلسیوس و تغذیه 3 ولت در حد 1 درصد تغییرات است.

اسیلاتور داخلی 32 مگاهرتز: برای استفاده از این اسیلاتور، باید ابتدا بوسیله خطوط برنامه عملیات فعال سازی آن انجام شود و در مرحله بعد بصورت مستقیم یا از طریق PLL به عنوان منبع کلاک سیستم انتخاب شود. دقت این اسیلاتور در دمای 25 درجه سلسیوس و تغذیه 3 ولت در حد 1 درصد تغییرات است.

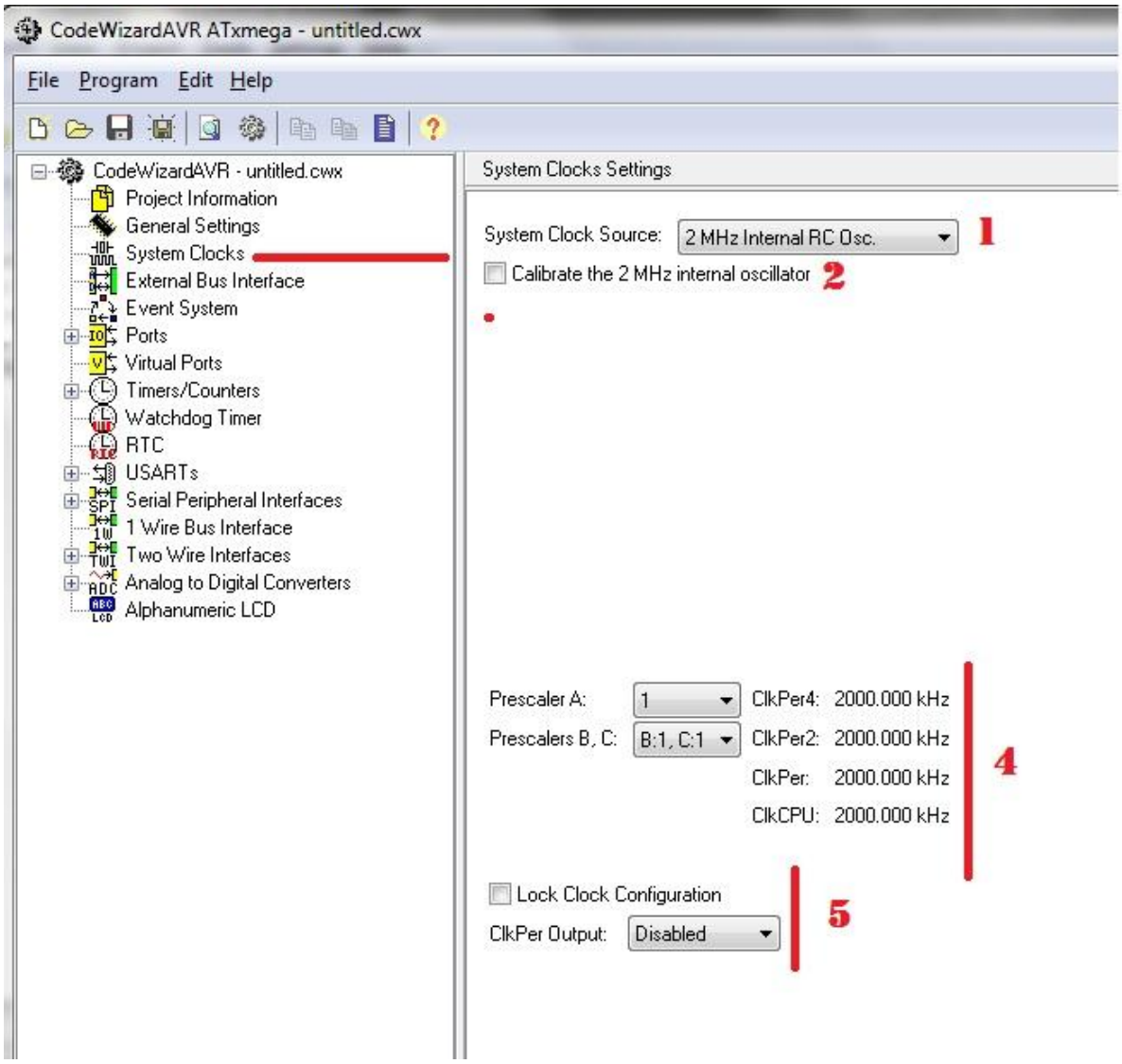
اسیلاتور داخلی 32.768 کیلوهرتز: این اسیلاتور بعد از فعال سازی می تواند به عنوان کلاک سیستم انتخاب شود و همچنین بعد از تقسیم شدن بر 32 برای اعمال به بخش RTC انتخاب شود. این اسیلاتور می تواند برای مکانیزم کالیبره کردن خودکار اسیلاتورهای 2 و 32 مگاهرتز داخلی بکار رود. دقت این اسیلاتور در دمای 25 درجه سلسیوس و تغذیه 3 ولت در حد 1 درصد تغییرات است. امکان کالیبره کردن مقدار خروجی این اسیلاتور بصورت نرم افزاری وجود دارد.

منابع کلاک خارجی: قابلیت اتصال کریستال یا رزوناتور و همچنین اعمال کلاک مستقیم از خارج وجود دارد. پین های XTAL1 و XTAL2 قابلیت اتصال کریستالی در محدوده 0.4 تا 16 مگاهرتز را دارند و به دو پین TOSC1 و TOSC2 می توان یک کریستال ساعت با فرکانس Hz32768 را متصل کرد. هر یک از این دو منبع به همراه کلاک خارجی اعمال شده به ورودی XTAL1 می توانند به عنوان منبع کلاک سیستم تعیین شوند.

PLL: برخی از منابع ذکر شده می توانند به عنوان ورودی به PLL تعیین شوند و مضربی از آن به عنوان کلاک سیستم انتخاب شود. اسیلاتور داخلی 2 مگاهرتز، اسیلاتور داخلی 32 مگاهرتز (تقسیم بر 4)،کلاک خارجی و خروجی اسیلاتور کریستالی فرکانس بالا (16-0.4 مگاهرتز) می توانند به عنوان ورودی PLL تعیین شوند. ضریب افزایش فرکانس PLL در محدوده 1 تا 31 برابر است. خروجی PLL نباید در فرکانسی کمتر از MHz10 تنظیم شود و عدد MHz200 به عنوان حد بالای آن ذکر شده است. برای تغییر ضریب PLL باید ابتدا این واحد غیر فعال شود و بعد از تغییر ضریب، مجددا فعال شود.

منوی system clocks:

توسط این قسمت می توانید منابع کلاک میکرو و واحد pll را تنظیم کنید.



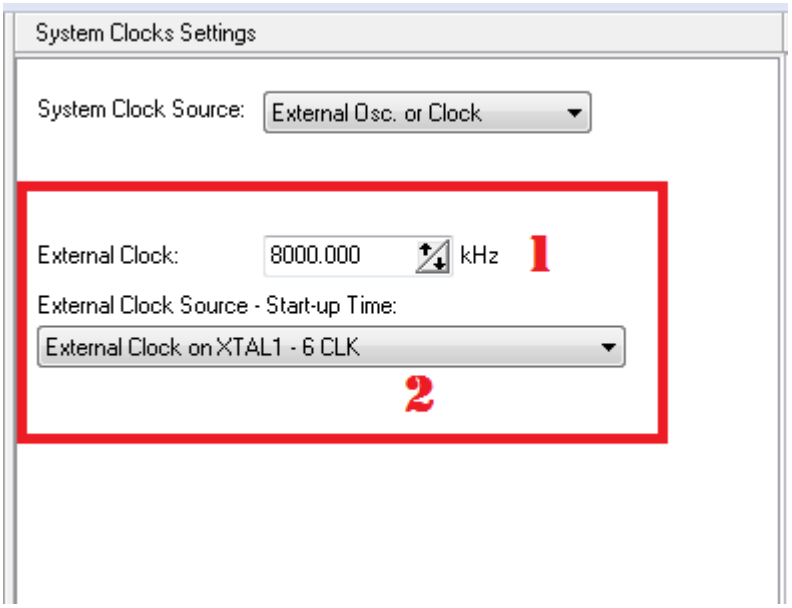
1-این قسمت برای انتخاب کلاک اصلی سیستم است و از بالا به پائین به ترتیب می توانید فرکانس های (2MHZ،32MHZ،32.768KHZ) داخلی یا EXTERNAL OSC (کریستال یا منبع کلاک خارجی)و یا PHASE LOCKED LOOP(PLL) را انتخاب کنید.

2- این اسیلاتور می تواند برای مکانیزم کالیبره کردن خودکار اسیلاتورهای 2 و 32 مگاهرتز داخلی بکار رود.

4-در این قسمت می توانیم با تنظیم PRESCALER A,B,C به کلاک های متفاوتی دست پیدا کنیم که از این کلاک برای وسائل جانبی متصل به میکرو،کلاک CPU یا قسمت های داخلی مانند واحد DMA,EVENT SYSTEM,EXTERNAL BUS,INTERRUPT به کار میرود.(دیتا شیت صفحه 78)

5-با فعال کردن این قسمت و انتخاب پایه مورد نظر در قسمت پائین ان(CLKPER OUTPUT) می توانید خروجی کلاک ایجاد شده توسط تنظیم PRESCALER های ذکر شده را روی پایه مورد نظر مشاهده کنید.که از این کلاک می توانید برای انواع وسائل جانبی متصل به میکرو استفاده کنید.

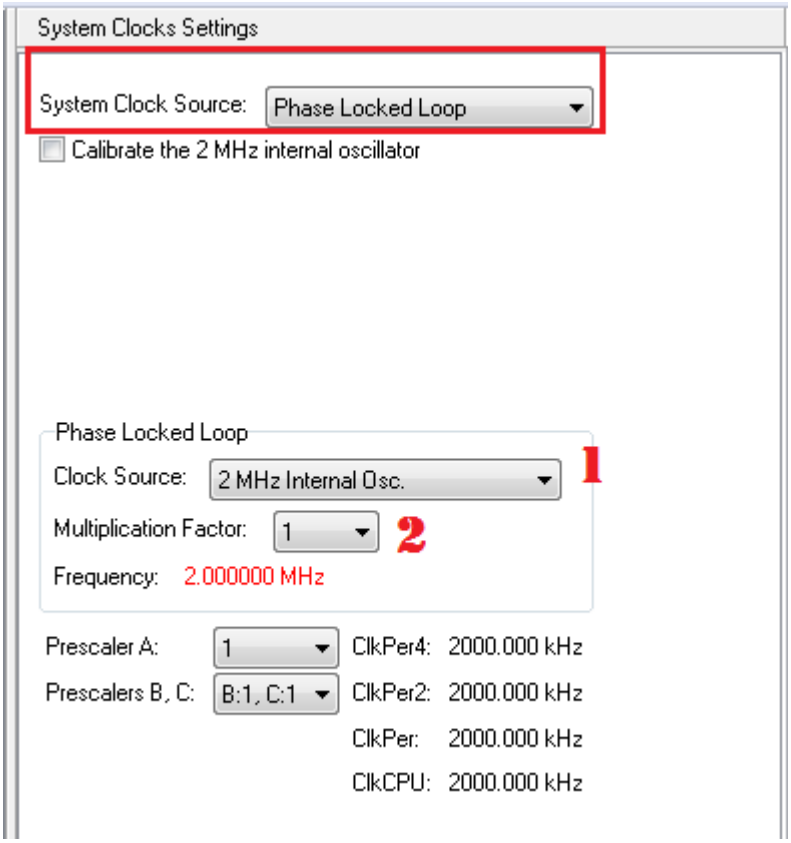
کریستال خارجی(external osc or clock):



با انتخاب این گزینه کادری در قسمت پائین ان باز می شود که به ترتیب می توانید در قسمت 1 مقدار کریستال متصل به میکرو و همچنین در قسمت 2 نوع منبع کلاک متصل شده به میکرو ،بازه ان و زمان startup انرا مشخص کنید.

واحد PLL:

با انتخاب قسمت PHASE LOCKED LOOP(PLL) وارد تنظیمات قسمت PLL می شویم.



1-در این قسمت منبع کلاک اصلی را انتخاب می کنیم.که می تواند 2MHZ،32MHZ داخلی و یا EXTERNAL CLOCK کریستال یا منبع کلاک خارجی باشد.

2-در این قسمت عدد مورد نظر را که مقدار کریستال در ان ظرب می شود را انتخاب می کنیم و عددی که در پائین ان به نمایش در می اید به عنوان فرکانس سیستم به کار می رود.(اگر کریستال 2MHZ داخلی باشد عدد مورد نظر در ان ضرب می شود و اگر 32MHZ داخلی باشدمقدار کریستال تقسیم بر 4 شده(32/4) بعد در عدد مورد نظر ظرب می شود.

توجه:

عدد به دست امده از واحد PLL نباید کمتر از 10MHZ باشد اعدادی که بعد از تنظیم PLL در قسمت FREQUENCY و PRESCALER ها به نمایش در می ایند اگر به رنگ قرمز در ایند عملکرد انها در ان فرکانس تظمین نمی شود و در حقیقت خطا به حساب می ایند.

پورت ها:

XMEGA – پورت ها

به دليل تغذيه حداكثر 3.6 ولت براي XMEGA ، حداكثر ولتاژ توليد شده به عنوان خروجي و حداكثر ولتاژ مجاز به عنوان ورودي بوسيله تغذيه محدود مي شود و اتصال به وسايل جانبي با تغذيه 5 ولت بايد با رعايت اين مورد انجام بپذيرد. نكته بعدي حالت هاي متنوع تر ورودي و خروجي

نسبت به AVR هاي معمول است. هر پين يك پورت به عنوان ورودي مي تواند در 4 وضعيت و به عنوان خروجي در 5 وضعيت مختلف از نظر عملكرد قرار داشته باشد.

از نظر ورودي حالت هاي زير قابل فعال شدن هستند:

1-High impedance

2-فعال شدنPull up

3-فعال شدنPull down

4-Bus keeper:كه به معناي فعال سازي خودكار Pull up يا Pull down براي حفظ وضعيت پورت متناسب با حالت خروجي آن است.

از نظر خروجي 5 وضعيت قابل فعال شدن هستند:

1-Totem pole:در اين حالت پين خروجي براي هردو وضعيت High و Low بصورت مناسب درايو مي شود.

2-Wired AND+Pull up:اين خروجي ها قابل وصل كردن به يكديگر هستند. خروجي هايي كه High هستند تاثيري در خروجي مشترك پين ها ندارند و خروجي هايي كه Low هستند خود را به وضعيت پين هاي خروجي تحميل مي كنند. اگر حتي يك خروجي Low باشد، پين خروجي صفر مي شود Pull up .بصورت داخلي فعال است.

3-Wired AND:مانند وضعيت قبل و بدون فعال بودن Pull up داخلي.

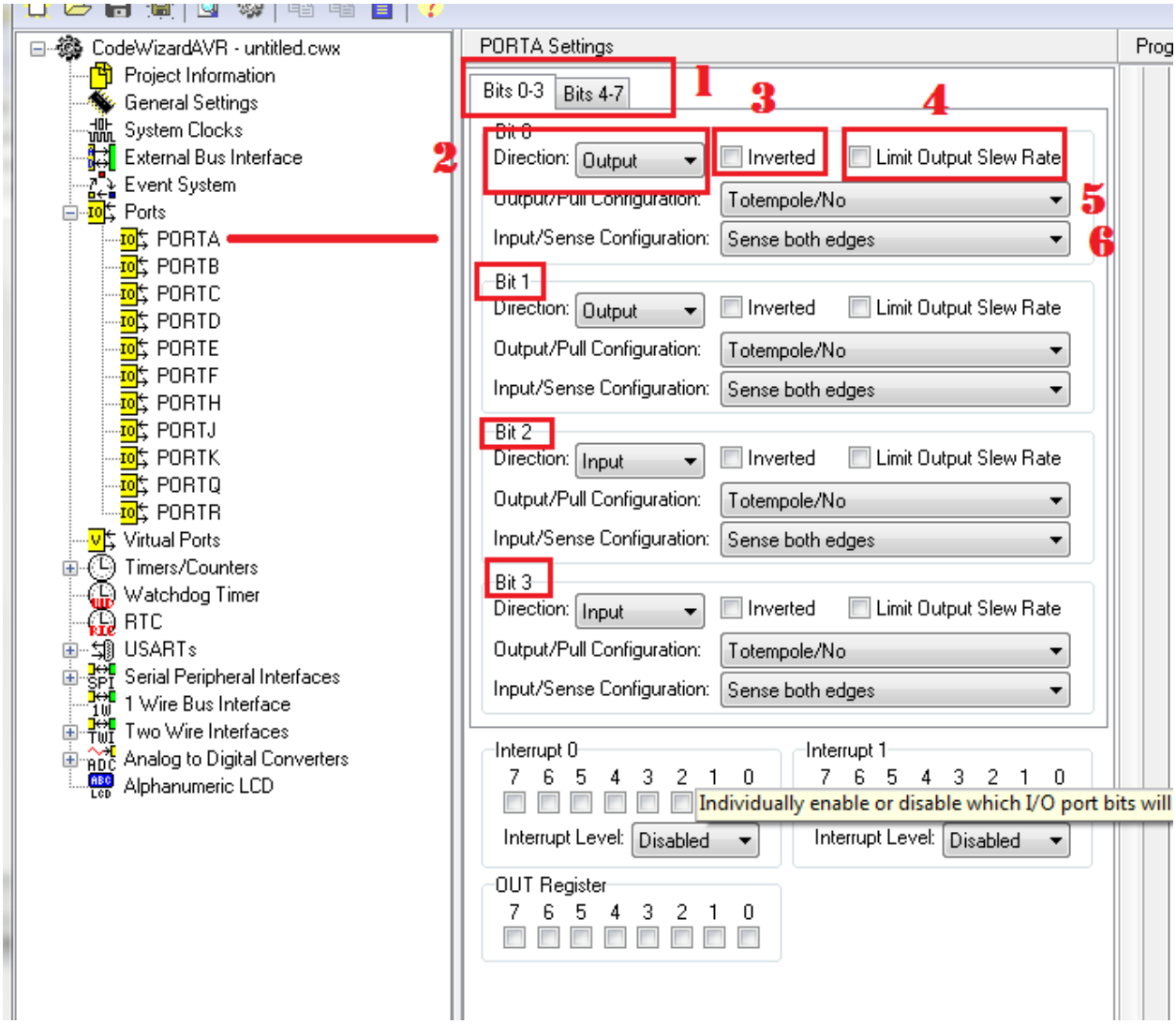
4-Wired OR + Pull down:اين خروجي ها قابل وصل كردن به يكديگر هستند.خروجي هايي كه Low هستند تاثيري در خروجي مشترك پين ها ندارند و خروجي هايي كه High هستند خود را به وضعيت پين هاي خروجي تحميل مي كنند. اگر حتي يك خروجي High باشد، پين خروجي يك مي شود Pull down .بصورت داخلي فعال است.

5- Wired OR:مانند وضعيت قبل و بدون فعال بودن Pull down داخلي.

مسئله ديگري كه در مورد پورت هاي XMEGA مي توان به آن اشاره كرد، امكان تعريف وقفه خارجي روي تمام پين هاي پورت هاست كه در مقايسه با AVR كه فقط پين هاي خاصي به اين امر اختصاص پيدا كرده، مزيت مهمي به شمار مي رود.

تنظیمات قسمت پورت در CODEWIZARD:

هر میکرو می تواند بسته به نوع ان تعداد پورت های متفاوتی داشته باشد که میکروکنترولر XMEGA128A1 دارای 11 پورت 8 بیتی می باشد،در زیر تنظیم یک بیت از پورت A این میکرو اموزش داده می شود که تنظیم بقیه بیت ها نیز به این صورت خواهد بود.



1-در این قسمت شماره بیتها را مشاهده می کنید که به صورت 4 تائی از هم جدا شده اند(3-0و7-4) و هر بیت به صورت جداگانه تنظیمات مربوط به خود را دارد.

2-در این قسمت جهت پورت تنظیم می گردد که ورودی باشد یا خروجی.

3-اگر این قسمت علامت گذاری شود خروجی پورت به صورت معکوس خواهد بود(اگر صفر بوده 1 می شود و اگر 1 بوده صفر می گردد)

4-توسط این گزینه می توانید خروجی SLAW RATE را محدود کنید(مثلا زمانی که طول می گشد خروجی از صفر به 3.6V برسد.البته در تغذیه 3.6)

5-توسط این قسمت می توان مقاومت های PULLUP,PULLDOWNو سایر حالات را فعال کرد.(رجوع شود به مطالب پورت های XMEGA-ذکر شده در بالا-)

6-در این قسمت حساسیت پورت ها در صورت استفاده از وقفه ها تنظیم می شود که از بالا به پائین به این صورت است.

الف-حساس به هر 2 لبه

ب-حساس به لبه بالا رونده

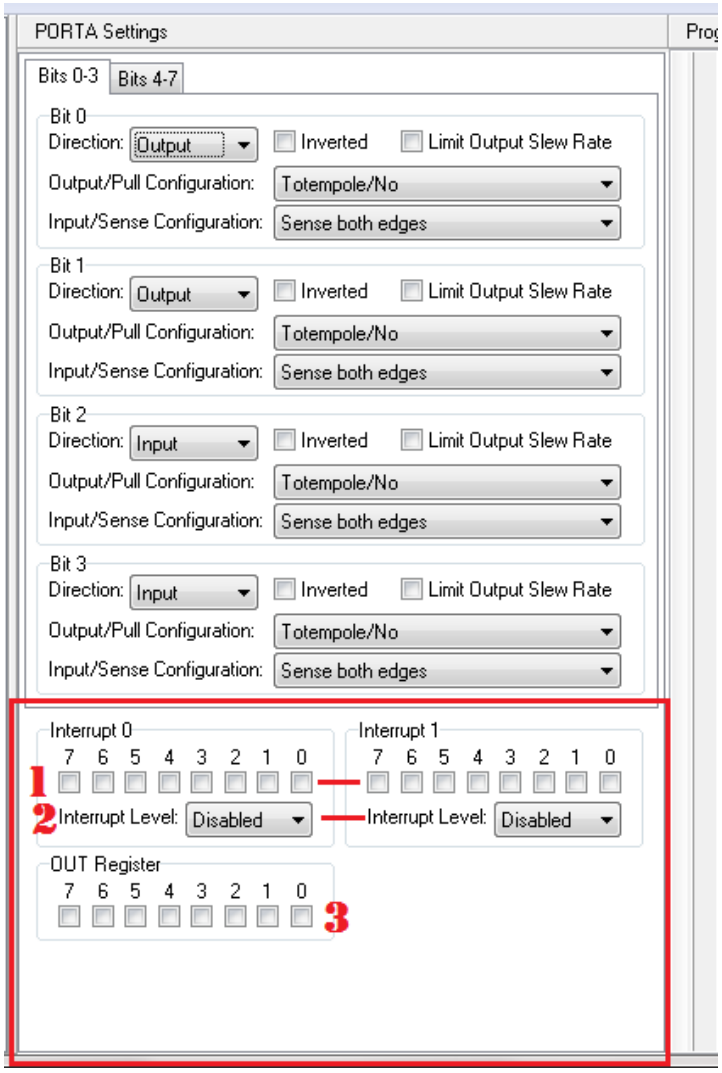
ج- حساس به لبه پائین رونده

د- حساس به سطح پائین

ه-غیر فعال کردن ورودی بافر

وقفه های پورت ها:

در صفحه تنظیم پورت ها و در قسمت پائین ان می توانیم یک سری تنظیمات از جمله تنظیم وقفه های مربوط به هر بیت(پایه) را انجام دهیم.



در عکس بالا مشاهده میکنید که برای هر پورت دو منبع وقفه داریم INTERRUPT0,INTERRUPT1 که عملکرد هر 2 یکسان بوده ودر عمل می توانید هر 2 را استفاده کنید.

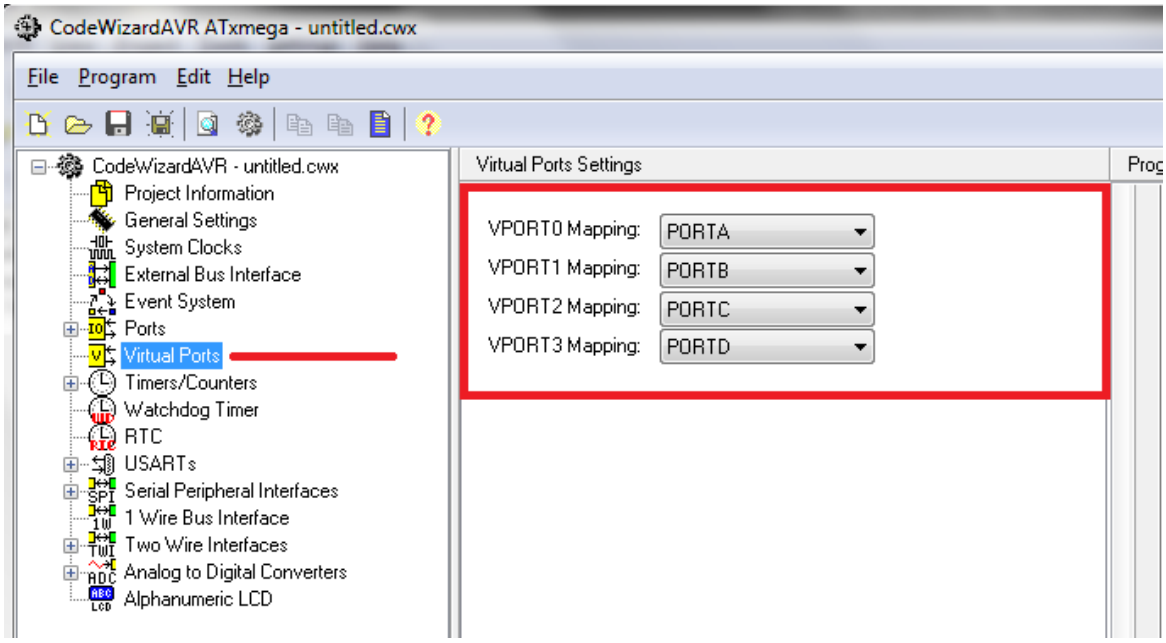
1-در این قسمت می توانیم وقفه مربوط به هر بیت(پایه)از پورت مورد نظر را فعال یا غیر فعال نمائیم.

2-در این قسمت می توانیم اولویت وقفه مورد نظر را تعیین کنیم.(در قسمت وقفه ها بیشتر بررسی خواهد شد)

3-در این قسمت می توانیم بیتی را که به عنوان خروجی قرار گرفته(یا تمام بیتها را) مقدار دهی اولیه نمائیم.

پورت های مجازی(VIRTUAL PORTS):

پورت های مجازی یکی دیگر از امکاناتی است که در میکرو کنترولر های XMEGA وجود دارد وعملکرد ان همانند پورت های واقعی میباشد یعنی هر عملی که روی انها انجام شود در بیت متناظر ان پورت نیز انجام می گردد.این عمل همچنین به شما اجازه می دهد تا از دستور العمل های خاص فضای I/O MEMORY برای تغییرات بیتی مانند ورودی یا خروجی کردن استفاده کنید که ادرس انها در فضای I/O MEMORY قرار خواهد گرفت.



در این میکرو کنترولر 4 پورت مجازی وجود دارد که می توانید به هر یک از انها یک پورت را MAP کنید.(پورت مورد نظر را در کادر روبروی هر VPORT انتخاب کنید)

انجام عملیات بیتی روی پورتها(ورودی ،خروجی .....):

براي تعيين وضعيت پورت ها در AVR هاي معمولي، به ازاي هرپورت 3 رجيستر وجود دارد. رجيستر DDRx براي تعيين جهت پورت، رجيستر PORTx براي تعيين مقدار خروجي و يا فعالسازي مقاومت Pullup در ورودي و رجيستر PINx براي خواندن مقدار واقعي پين در

نظر گرفته شده اند. اما در XMEGA نام وعملکرد رجیسترها بصورت متفاوتي است که در اینجا توضیح داده مي شود. براي تعیین جهت هر پورت از رجیستري با نام PORTx.DIR استفاده مي شود(مثلا PORTA.DIR). همانند AVR نوشتن یک در هر بیت از این رجیستر منجر به خروجي شدن پین متناظر آن در پورت مي شود، اما وضعیت فعالسازي مقاومت Pullup ربطي به این رجیستر ندارد.به غیر از این رجیستر، تعداد 3 رجیستر دیگر در همین رابطه پیش بیني شده که عملکرد آنها در AVR مشابهي ندارد و براي بالا بردن سرعت انجام عملیات در نظر گرفته شده اند.PORTx.DIRSET رجیستري است که نوشتن یک در هر بیت آن موجب خروجي شدن پین هاي متناظر در آن پورت مي شود، اما وضعیت سایر بیت ها بدون تغییر باقي مي مانند. مثلا با اجراي دستور PORTA.DIRSET=0x0F; در زبان C فقط 4 بیت پائین پورت خروجي مي شوند و 4 بیت بالا در هر وضعیتي از نظر ورودي یا خروجي که بودند، باقي مي مانند. در همین رابطه دو رجیستر دیگر با نامهاي PORTx.DIRCLR و PORTx.DIRTGL وجود دارند که نوشتن یک در بیت هاي رجیستر اول منجر به ورودي شدن بیتهاي متناظر و بدون تغییر ماندن جهت سایر بیتها مي شود و براي رجیستر دوم، نوشتن یک منجر بهToggle شدن وضعیت بیت ها از نظر جهت مي شود.

براي تغییر مقدار خروجي هم 4 رجیستر براي هر پورت به نام هاي PORTx.OUT و PORTx.OUTSET و PORTx.OUTCLR و PORTx.OUTTGL وجود دارند که رجیستر اول منجر به تغییر مقادیر کلیه بیت هاي پورت مي شود و در 3 رجیستر بعدي فقط بیت هایي که در آن یک نوشته شده تغییر مي کنند و سایر بیت ها بدون تغییر باقي مي مانند.

براي خواندن وضعیت هر پورت هم رجیستر PORTx.IN وجود دارد که عملکردي مشابه رجیستر PINx در AVR هاي عادي دارد.با این توضیهات چند مثال از عملکرد پورتها در زیر ذکر می شود.

(تمام مثالهال زیر در فرکانس 2MHZ نوشته شده اند.قبل از نوشتن برنامه در CODEVISION تنظیمات مربوط به پورت مورد نظر را در قسمتCODEWIZARD انجام دهید(پایه های پورت مورد نظر را بسته به موارد نیاز ورودی یا خروجی یا ..... تنظیم کنید))

مثال1:برنامه ای بنویسید که هر 500MS وضعیت PORTA را برعکس کند.

در صورت استفاده از دستورات تاخیر کتابخانه delay را باید در قسمت include معرفی کنید.(#include “delay.h”)

```
Void main(void){

PORTA.DIR=0XFF;

While(1){

PORTA.OUTTGL=0XFF;

Delay_ms(500);

  }

}
```

مثال2:برنامه ای بنویسید که به ترتیب POATA,PORTB را 1ثانیه روشن و بعد خاموش کند.

```
Void main(void){

PORTA.DIR=0XFF;

PORTAB.DIR=0XFF;
```

```
While(1){

PORTA.OUT=0XFF;

Delay_ms(1000);

PORTA.OUTCLR=0XFF;

Delay_ms(1000);

PORTB.OUT=0XFF;

Delay_ms(1000);

PORTB.OUTCLR=0XFF;

  }

}
```

مثال4:برنامه ای بنوسیسد که با استفاده از پورت های مجازی 4بیت (پایه) از PORTK را روشن و خاموش کند.(در این مثال PORTK به VPORT0 (پورت مجازی اول MAP شده است.))

(بعد از اینکه در قسمت پورت های مجازی در CODEWIZARD پورتی را که می خواهید MAP شود انتخاب کردید باید در تابع مورد نظر که تنظیمات VPORT که در انجا انجام می شود پورت MAP شده مورد نظر را خروجی یا ورودی ،بسته به موارد نیاز پیکر بندی کنید)

```
// Virtual Ports initialization
void vports_init(void)
{
// PORTK mapped to VPORT0
// PORTB mapped to VPORT1
PORTCFG.VPCTRLA=PORTCFG_VP1MAP_PORTB_gc | PORTCFG_VP0MAP_PORTK_gc;
// PORTC mapped to VPORT2
// PORTD mapped to VPORT3
PORTCFG.VPCTRLB=PORTCFG_VP3MAP_PORTD_gc | PORTCFG_VP2MAP_PORTC_gc;
VPORT0.DIR=0XFF;
}
```

طبق مثال نوشته شده PORTK به VPORT0 پورت مجازی اول MAP شده پس برای خروجی کردن پورت K به این صورت داخل دستور زیر را به قسمت تنظیمات این تابع اضافه می کنیم. VPORT0.DIR=0XFF;

```
Void main(void) {

While(1) {

VPORT0.OUT=0XF;

delay_ms(500);
```

```
VPORT0.OUT=0XF0;

delay_ms(500);

  }

}
```

مثال 5:برنامه ای بنویسید که با فشرن کلید متصل به PORTF.0 یک LED متصل به PORTK.0 را روشن کند.

(در CODEWIZARD پایه مورد نظر از پورت Fرا ورودی کرده و مقاومت PULL/UP ان را فعال کنید همچنین پایه مورد نظر از پورت K را خروجی تنظیم نمائید)

```
Void main(void) {

PORTF.DIR=0;

PORK.DIR=0XFF;

While(1) {

If((PORTF.IN &(1<<0))==0)

PORTK.OUTSET=1;

  }

}
```

پایان قسمت اول

منابع:

1–**مهندس اوژن کی نژاد**

2–**سایت ATMEL**

3–**سایت AVRFREAKS**

P-A