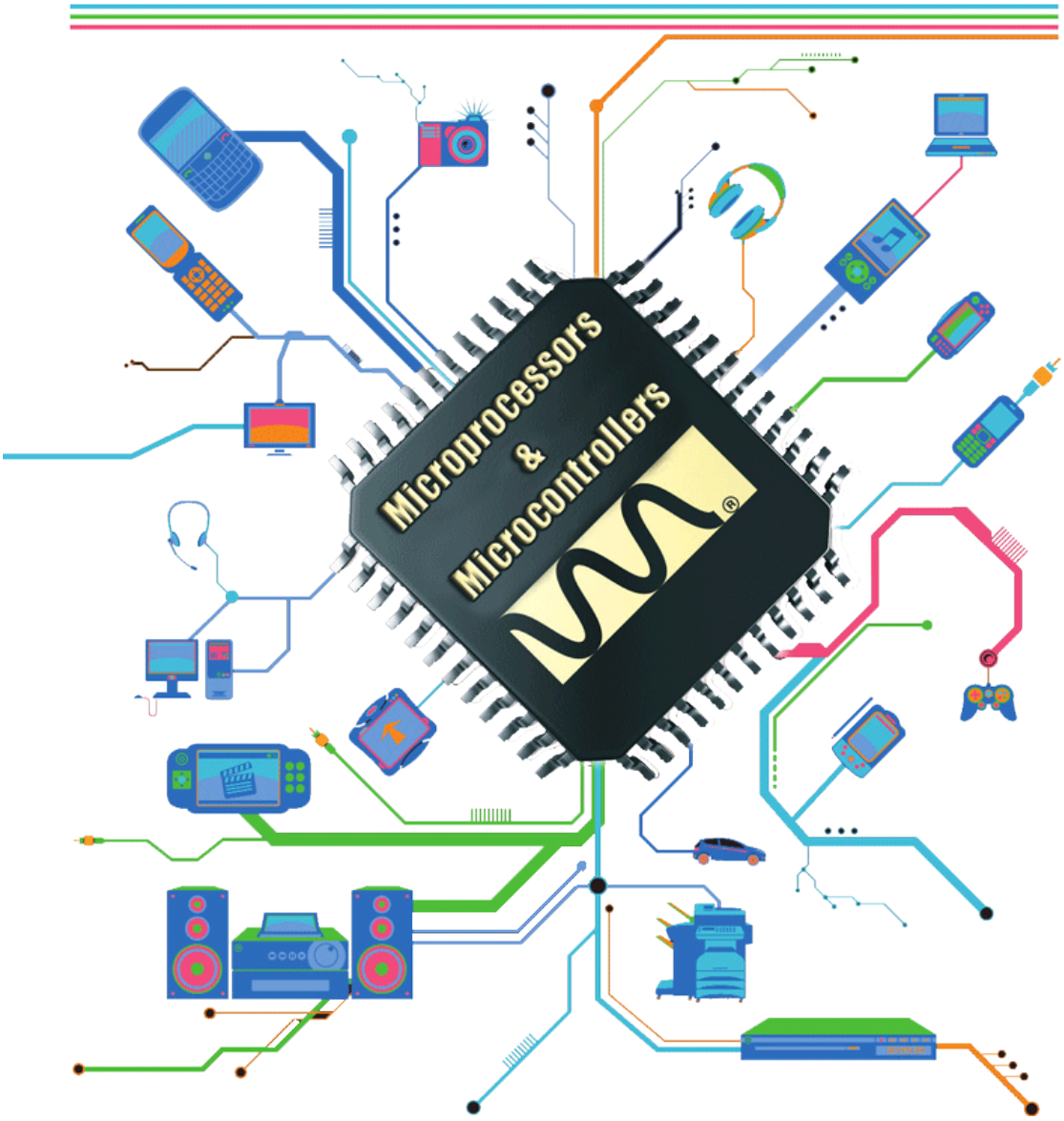
جامعـــــــــــة حلـــــــــــب
كلية الهندسة الكهربائية والإلكترونية
قسـم هندسـة التحكـم والأتمتـة

﴿ الجلسات العملية لمادة المعالجات والمتحكمات المصغرة ﴾

Microprocessors & Microcontrollers Lab Sessions

السنة الثالثة | قسم اتصالات

﴿ الجلسة العملية الثانية ﴾



م. وليد بليد

Copyright © 2012 Walid Balid - All rights reserved.

Wednesday, March 07, 2012

﴿الجلسة العملية الثانية﴾

نظرة عامة (Overview):

هذه الجلسة تقدم تصنيفاً للغات برمجة الأنظمة المدمجة ومعايير اختيار الحلول التكنلوجية. ثم نظرة عامة عن الأدوات البرمجية ولوحة التطوير التي ستستخدم خلال الجلسات العملية ومنهجية تنفيذ التجارب. ثم مدخل إلى متحكمات AVR وعائلاتها والبنية الداخلية وتنظيم الذاكرة، ثم نظرة عامة على ميزات المتحكم ATmega32A.

1-2 تمهيد (Preface):

مما لا شك فيه أن لغة التجميع (Assembly) هي أكثر فعالية من غيرها من اللغات ذات المستوى الأعلى عندما يتعلق الأمر بالتعامل المباشر مع وظائف وسلوك المعالج أو المتحكم المصغر، ولكن هذا من جانب آخر يحتاج إلى مستوى عالٍ من الخبرة في بنية الكيان الصلب للمعالج لتوظيف لغات التجميع بشكل فعال، كما أنه سيحتاج إلى أضعاف مضاعفة من الوقت لبناء تطبيق محدد؛ السبب الذي يجعل مطوري البرامج الحاسوبية يعانون من نقص الخبرة حول تفاصيل وتعقيدات تطوير الكيان الصلب؛ مما يحد من مقدرتهم على تصميم الانظمة المدمجة (**ESs**).

للاستفادة من مهارات مطوري البرمجيات الحاسوبية، والحد من طابع تعقيدات برمجة الكيان الصلب بلغة التجميع، قامت شركات الـ"EDA" (Electronic Design Automation) بتطوير بيئات وأدوات برمجية تستخدم لغات عالية المستوى مثل ال:-C/C++ القياسية أو لغة الBasic أو لغة الPascal لتطوير وبرمجة الأنظمة المدمجة. هذه الأدوات البرمجية الجديدة تقوم على تحويل البرنامج من لغة عالية المستوى إلى برنامج بلغة التجميع منخفض المستوى وتدعى في أغلب الأحيان بـMappers أو Compilers. تمتلك هذه الأدوات المقدرة على تطوير وفحص وتتبع أخطاء البرامج التي هي مشابهة جدا لبيئات تطوير البرمجيات.

الجذور الأولى لنشأة لغات البرمجة عالية المستوى كانت مع إشراقة فجر عصر الحوسبة الحديثة في منتصف الخمسينيات من القرن الماضي، حيث كان فريق صغير من الباحثين في شركة IBM قد قرر إيجاد بديل آخر لاستخدام لغة التجميع منخفضة المستوى (Assembly) في برمجة الحاسب IBM-704، وكانت النتيجة ظهور لغة Fortran – شكلٌ آخر من أشكال لغات البرمجة أكثر قابلية للقراءة والفهم – والتي تهدف في الأساس إلى تسريع عمليات تطوير البرامج المختلفة.

لقد انتاب المجتمع الهندسي في البدء بعضُ الشكوك في كون هذه الطريقة الجديدة قادرة على التفوق على البرامج المكتوبة يدوياً بلغة التجميع، ولكن سرعان ما ثبت أن البرامج المكتوبة بلغة Fortran قادرةٌ على العمل تقريباً بنفس فعالية تلك المكتوبة بلغة التجميع؛ وفي نفس الوقت، استطاعت لغة Fortran تقليص عدد التعليمات البرمجية المستخدمة لبناء برنامج ما بحوالي عشرين مرة، وهذا ما جعلها

تعتبر أولى لغات البرمجة عالية المستوى، ولم يكن من المفاجئ أن لغة Fortran قد حصلت بسرعة كبيرة على رضى وقبول المجتمع العلمي في ذاك الوقت وحتى وقت متأخر. بعد نصف قرن، ما زلنا نستطيع استخلاص الكثير من العِبَر الهامة من هذه القصة وهي:

- ✓ لأكثر من خمسين عاماً، حاول المهندسون ابتكار طرق أسهل وأسرع لحل المشكلات باستخدام البرمجة الحاسوبية.
- ✓ لغات البرمجة التي اختارها المهندسون لتلبية متطلباتهم اتجهت نحو مستويات أعلى (High-level of Abstraction).

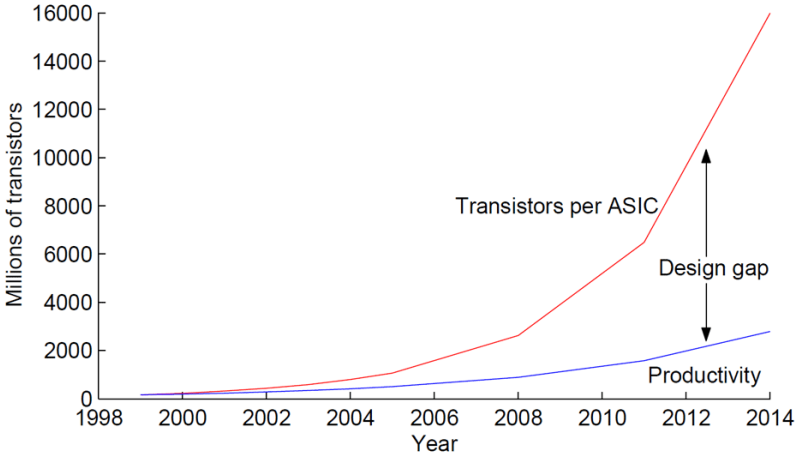
2-2 تصنيف لغات برمجة الأنظمة المدمجة (ESs. Programming Languages):

تصنف لغات برمجة الأنظمة المدمجة وفق أربع مجموعات رئيسية، وهي:

1. لغات البرمجة منخفضة المستوى – لغة التجميع (Low-level Programming Languages, Assembly).
2. لغات البرمجة عالية المستوى – الوظيفية (High-level F. Programming Language) مثل: Basic, Pascal.
3. لغات البرمجة عالية المستوى – الإجرائية (High-level P. Programming Languages) مثل: Embedded C/C++.
4. لغات البرمجة الرسومية (Graphical Programming Languages) مثل: LabVIEW.

3-2 نحو لغات برمجية للأنظمة المدمجة عالية المستوى (Toward High-level ESs. Programming Languages):

إن الازدياد المتسارع في تعقيد بنية شرائح الدارات المتكاملة وفقاً لقانون "Moore" يؤدي إلى نشوء فجوة كبيرة بين عدد الكتل المنطقية التي يمكن إنشاؤها على مساحة محددة من شريحة سيليكونية وبين إمكانية مهندسي التصميم على استثمار هذه الشريحة المتكاملة برمجياً، وهذا ما يدعى الآن بـ"Design Gap". الشكل1 يبين العلاقة بين الإنتاجية التصميمية لمهندسي تصميم الشرائح المتكاملة وبين عدد الترانزستورات التي يمكن أن توضع على شريحة سيليكونية.



الشكل1 العلاقة بين ازدياد تعقيد الدارات المتكاملة ومقدرة المصممين على تطويرها

إن نشوء هذه الفجوة يقود إلى الحاجة في البحث عن طرائق ووسائل لبرمجة الكيان الصلب بمستويات برمجية أعلى وهو ما يعرف اصطلاحاً بـ"high-level abstraction". على الرغم من أن التصميم الذي يمكن إنشاؤه من مستوى برمجي أعلى قد يكون أقل كفاءة مقارنةً مع التصميم الذي يتم إنشاؤه مباشرة باستخدام لغات التجميع، إلا أن هذا الأمر أقل أهمية بكثير من مسائل التعقيد البرمجي والجهود الكبيرة

والزمن المصروف للبرمجة بلغات التجميع. على كل حال فإنه مؤخراً تمّ بذل العديد من الجهود تهدف لإيجاد أدوات برمجية بلغات عالية المستوى تهدف إلى الوصول لتصميم أمثلي. هناك مسألة أخرى تتعلق باستخدام لغات عالية المستوى في برمجة الأنظمة المدمجة وهي أنه لكي تكون ناجحاً في الاقتصاد العالمي اليوم، فإن مسألة وصول المنتج إلى السوق يجب أن تتم بشكل أسرع من السابق، وبالتالي فإن دورة تصميم المنتج يجب أن تكون أقصر ما يمكن.

4-2 معايير اختيار الحلول التكنلوجية (Technology Solutions Selection Criteria):

كما أن هناك عوامل مؤثرة في تصميم الأنظمة المدمجة (الفقرة 5-1)، فإنه في مشاريع الأنظمة المدمجة عموماً هناك معايير أساسية تعتمدها الشركات في اختيار الحلول التكنلوجية والعناصر الأساسية في المنتجات الإلكترونية – من أهم العناصر الأساسية هي الشريحة المدمجة؛ متحكم مصغر (MCU)، معالج مصغر (MPU)، معالج إشارة رقمية (DSP)، مصفوفة بوابات منطقية قابلة للبرمجة حقلياً (FPGA)، دارات متكاملة ذات تطبيقات خاصة (ASIC) – في منتجاتها والتي منها: تكلفة الشريحة، مستوى المرونة في بيئة التطوير، مستوى الدعم الفني من قبل الشركة المصنعة، مستوى الشركة المصنعة تجارياً. إن جميع هذه النقاط ضرورية جداً في تحديد أفق التطوير المستقبلي للمشاريع، حيث لا يمكن الاعتماد على الشركات المنطلقة حديثاً (Startups) عند اختيار المعالجات والشرائح الرقمية القابلة للبرمجة. الفقرات التالية تعالج هذه المعايير والاعتبارات.

5-2 معايير اختيار الشركات المصنعة (Manufacturer Selection Criteria):

يوجد العديد من المعايير والعوامل الهامة في اختيار الشركة المصنعة للحلول التكنلوجية نوردها فيما يلي:

- ✓ توفير الأدوات والحلول البرمجية (Tools).
- ✓ الريادة في التكنولوجيا (Technology leadership).
- ✓ تزويد الوحدات البرمجية المتوافقة مع الحلول البرمجية (IP offerings).
- ✓ تزويد منتجات بمواصفات ومزايا إبداعية (Innovative product features).
- ✓ تطوير أجيال المنتجات بشكل متجدد (Solid roadmaps).
- ✓ دعم وتصنيع عائلات الشرائح لفترات طويلة جداً (Longevity of parts).
- ✓ توفر العديد من عائلات الشرائح بميزات عديدة (Multiple families).
- ✓ الدعم الفني والتقني (Support).

6-2 عوامل اختيار الشريحة المناسبة للتطبيق (Chip Selection Factors):

بما أن الشركات المصنعة الرائدة توفر العديد من العائلات والشرائح التي تتفاوت في ميزتها وأدائها وسعرها، فإنه من الضروري اختيار الشريحة المناسبة للتطبيق من خلال تحديد العوامل المطلوبة وهي:

- ✓ السعر (Cost).

- ✓ الحجم (Size).
- ✓ الطاقة (Power).
- ✓ السرعة (Speed).
- ✓ عدد أقطاب الدخل والخرج (I/O Count).
- ✓ المصادر المنطقية الأساسية على الشريحة (Logic fabric resources).
- ✓ مصادر إدارة تردد عمل الشريحة (Clock management resources).
- ✓ مصادر الذاكرة (Memory resources).
- ✓ الشكل الفيزيائي للشريحة (Packaging).
- ✓ إمكانية وجود بدائل للشريحة (Common footprint component migration options).
- ✓ متطلبات الربط مع المحيطيات (Interface requirements).
- ✓ ميزات أدوات التصميم وانتشارها (Design tool features and familiarity).

خلال دراستنا هذه وقع الاختيار على شركة ATMEL الرائدة في مجال تصنيع المتحكمات المصغرة حيث تعتبر من أقوى الشركات عالمياً في مجال تطوير وتصنيع متحكمات 8-bit وفقاً لتقرير Gartner، كما أن تنتج طيفاً واسعاً من متحكمات 8-bit بأداء عالٍ وميزات كبيرة جداً وكل ذلك بسعر منخفض. أضف إلى ذلك انتشارها الواسع جداً (حتى في أسواقنا المحلية المتواضعة) وكثرة المصادر التعليمية المتوفرة على الشبكة..

7-2 لغات برمجة المتحكمات المصغرة (Microcontrollers Programming Languages):

يعتمد عمل المتحكم المصغر على مبدأ أساسي وهو تنفيذ مجموعة التعليمات الموجودة بداخله في ذاكرة البرنامج, إذ يقوم بتنفيذ التعليمات بشكل تتابعي (تعليمة تلو الأخرى) إلى أن يصل إلى نهاية البرنامج، وعندها يعود إلى التعليمة الأولى في بداية البرنامج ليبدأ بتنفيذ دورة أخرى من البرنامج وهكذا...

تختلف اللغات المستخدمة في كتابة برنامج المتحكم وتتفاوت في مقدار صعوبتها وتعقيدها, وسابقاً كانت تتم برمجة المتحكمات بشكل أساسي باستخدام لغة التجميع (Assembly)، حيث يمتلك كل نوع من المتحكمات لغة تجميع خاصة به تتعلق بنواة المعالج, حتى أن متحكمات نفس الشركة تختلف في لغة التجميع الخاصة بها من نوعٍ إلى آخر؛ فمتحكمات شركة ATMEL – على سبيل المثال – ذات نواة AT89xxxx تمتلك لغة تجميع تختلف عن لغة التجميع الخاصة بنواة متحكمات AT90xxxx. ومن المعروف أن لغة التجميع لغة اختصاصية وغير مرنة، كما أنها صعبة التدقيق والمراجعة والتطوير... فكتابة برنامج ما باستخدام لغة التجميع سيحتاج إلى وقت طويل وخبرة كبيرة من قبل المبرمج.

مع التطور التكنولوجي الكبير في صناعة الأنظمة المدمجة عموماً والمتحكمات المصغرة خصوصاً تنبهت الشركات إلى أن استمرار استخدام لغة التجميع يحد من سهولة استخدام هذه المتحكمات، وبالتالي تبقى محصورة ضمن فئة معينة من المهندسين المختصين، لذلك تم البحث عن طرق أسهل لبرمجة المتحكمات، مما أدى إلى ظهور لغات البرمجة عالية المستوى (High Level Languages) مثل: لغة C/C++ ولغة BASIC ولغة PASCAL وغيرها... في هذه اللغات يقوم المبرمج بكتابة البرنامج بإحدى لغات البرمجة عالية المستوى ويقوم مترجم خاص (Compiler) بتحويل هذا البرنامج إلى البرنامج المقابل له في لغة التجميع الخاصة بالمتحكم المصغر, وبالتالي لا حاجة إلى تعلم لغة التجميع كلما احتاج المبرمج استخدام عائلة جديدة من المتحكمات. من أشهر مترجمات متحكمات AVR نذكر:

اسم المترجم	اللغة	موقع الشركة
BASCOM-AVR	Basic	http://www.mcselec.com/
AVR-Studio	C/C++	http://www.atmel.com/
CodeVisionAVR	C	http://www.hpinfotech.ro/
Win-AVR	C++	http://winavr.sourceforge.net/
ImageCraft ICC-AVR	C	http://www.imagecraft.com/
MikroPascal For AVR	Pascal	http://www.mikroe.com/
MikroBasic For AVR	Basic	http://www.mikroe.com/

وبالتالي وباستخدام لغات برمجة عالية المستوى انكسر حاجز الاختصاص التي تفرضه لغة التجميع وأصبحت المتحكمات في متناول الجميع ممن لديهم خبرة متواضعة في لغات البرمجة ومعرفة كافية في بنية المتحكمات, كما أن وجود المكتبات المختلفة في هذه المترجمات جعلت عملية برمجة المتحكمات عملية ممتعة وسلسة ولا تحتاج إلى الكثير من الوقت في دراسة متطلبات الكيان الصلب. فيما يلي مقارنة سريعة لبعض الميزات الأساسية للّغات عالية المستوى مع ما يقابلها من اللغات المنخفضة المستوى.

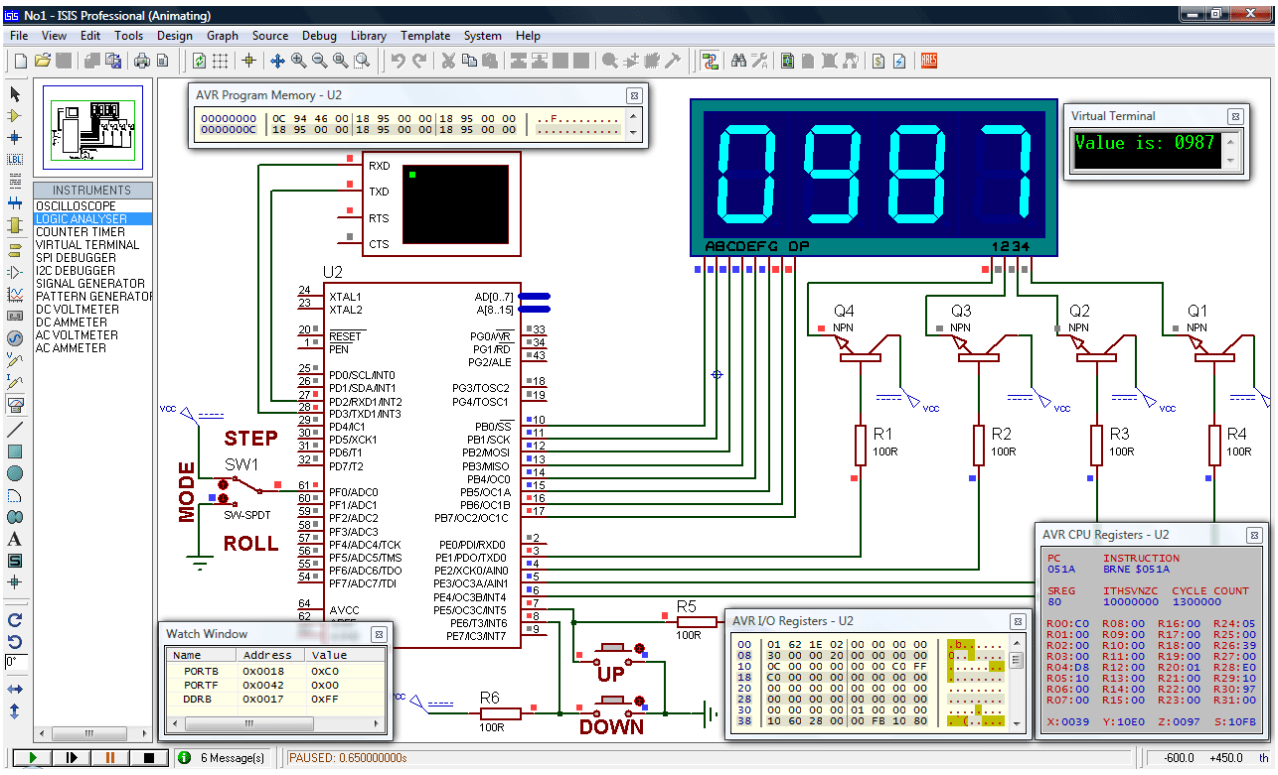
برامج مكتوبة بلغات برمجة منخفضة المستوى (Assembly)	برامج مكتوبة بلغات برمجة عالية المستوى (Basic, Pascal, C)
– تحتاج إلى وقت كبير لتعلمها	– يمكن تعلمها بوقت قصير جداً
– كتابتها تحتاج إلى خبرة كبيرة وأشخاص مختصين	– كتابتها لا تحتاج إلى تلك الخبرة وهي متاحة للجميع
– صعوبة في كتابة البرامج وتنقيحها ومراجعتها وتطويرها	– سلاسة في كتابة البرامج وتنقيحها وتطويرها
– برامج طويلة تتألف من عدد كبير جداً من التعليمات	– برامج أقصر وذات تعليمات أقل وأبسط
– برامج صعبة الكتابة القراءة والفهم	– برامج سهلة الكتابة وواضحة القراءة وتفهم بسهولة
– تستهلك مساحة أصغر من ذاكرة البرنامج	– تستهلك مساحة أكبر من ذاكرة البرنامج
– عدد محدود جداً من التعليمات الأساسية	– عدد كبير جداً من التعليمات الوظيفية تسهل البرمجة

8-2 حول منهجية تنفيذ التجارب العملية (About the Laboratory Sessions):

في هذا المنهاج العملي التطبيقي سوف نعالج بشكل خاص برمجة المتحكمات المصغرة من العائلة AVR باستخدام لغة عالية المستوى وهي لغة BASCI والتي تعتمدها بيئة التطوير Bascom-AVR (IDE). كذلك سنقوم بمحاكاة جميع الأمثلة والتطبيقات في بيئة المحاكاة LabCenter Proteus الذي يعد من أقوى البرامج التي تحاكي عمل المعالجات. سنقوم بعدها بتنفيذ التجارب عملياً على لوحة التطوير mini-Phoenix المعدة خصيصاً لهذا المختبر والتي تم تصميمها بحيث ترتقي بالمتعلم من مستوى مبتدئ إلى مستوى متقدم متضمنةً أكثر من 25 تجربة تشمل جميع الوظائف الأساسية للمتحكمات بالإضافة إلى وظائف متقدمة أخرى.

9-2 بيئة المحاكاة (The Simulation Environment):

في مرحلة التصميم وقبل التطبيق العملي للأمثلة والتمارين على لوحة التطوير، فسيتم محاكاة برامج الوحدات المحيطية للوحة التطوير في البرنامج PROTEUS والذي هو عبارة عن بيئة مخصصة لأغراض محاكاة الأنظمة الرقمية والمعالجات المصغرة.



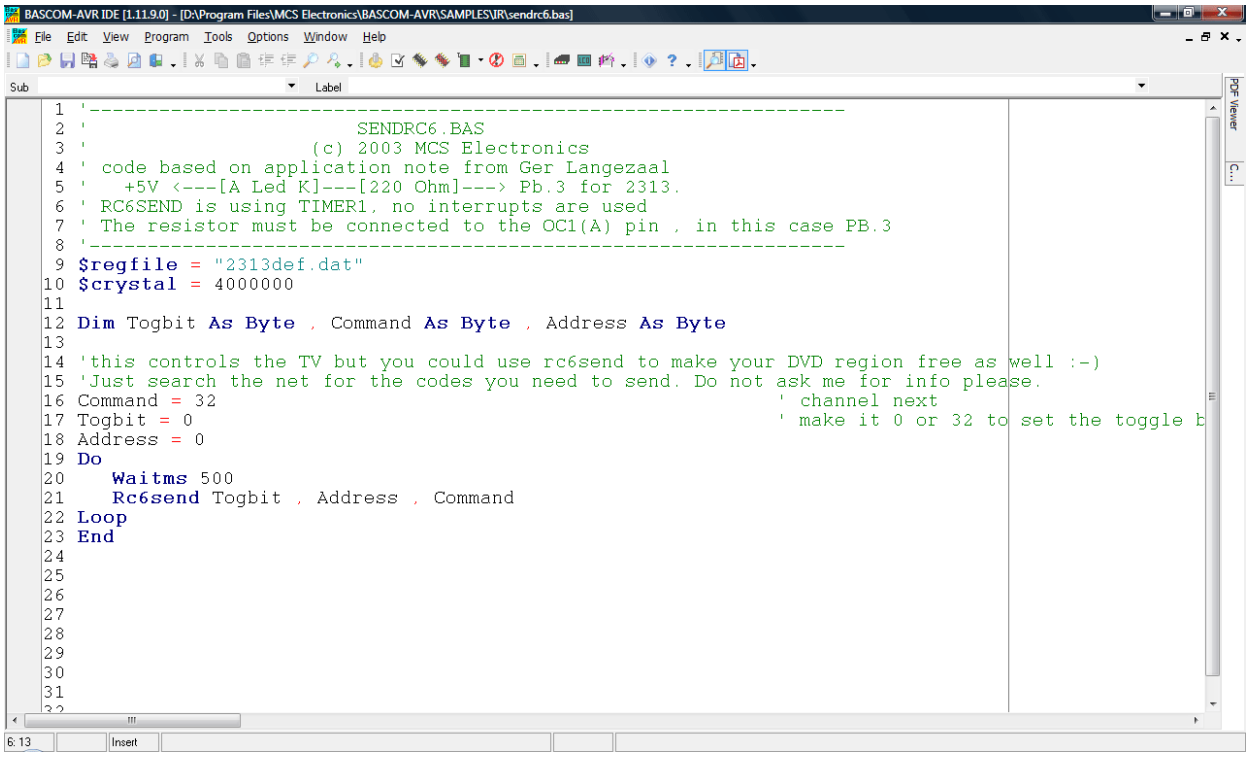
الشكل2 استخدام بيئة المحاكاة PROTEUS لتحليل دارة عداد تصاعدي تنازلي قابل للضبط

يعتبر برنامج PROTEUS من أقوى برامج المحاكاة للمتحكمات المصغرة، وهو يملك العديد من المكتبات التي تغطي جميع أنواع المحيطيات التي يمكن وصلها مع المتحكم المصغر بالإضافة إلى أدوات القياس العديدة، وسوف نستخدم هذا البرنامج لمحاكاة جميع التجاب التي سوف نتطرق إليها لاحقاً، كما سيتم استخدامه بشكل أساسي في المختبر الافتراضي.

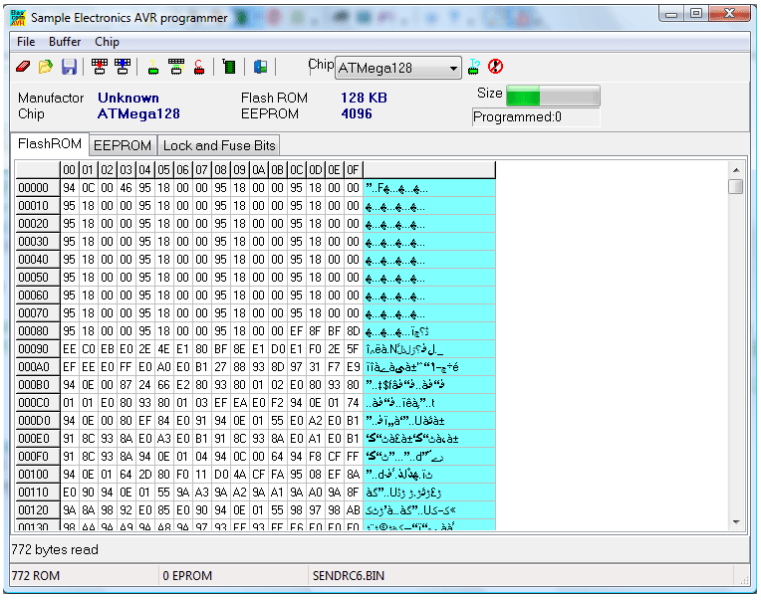
10-2 بيئة التطوير Bascom-AVR (The Integrated Development Environment):

تعتبر بيئة التطوير Bascom-AVR (IDE) من أشهر وأقوى بيئات التطوير البرمجية التي تستخدم لغة عالية المستوى وهي لغة BASCI لبرمجة المتحكمات المصغرة من العائلة AVR. تمتلك هذه البيئة واجهات تطبيقات متعددة وهي:

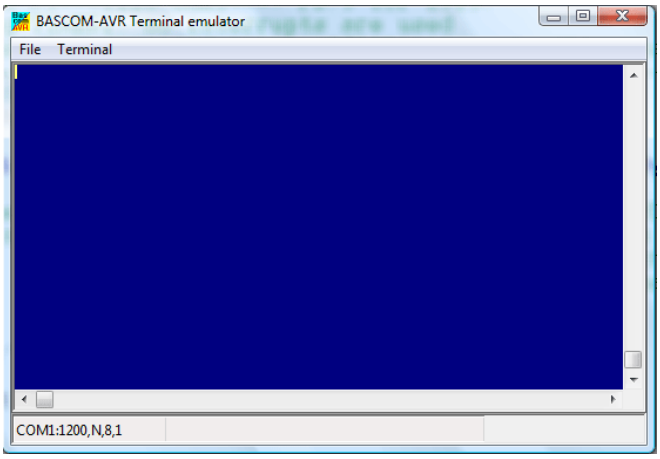
1. الواجهة البرمجية الرئيسية: وهي محرر التعليمات والأوامر البرمجية.



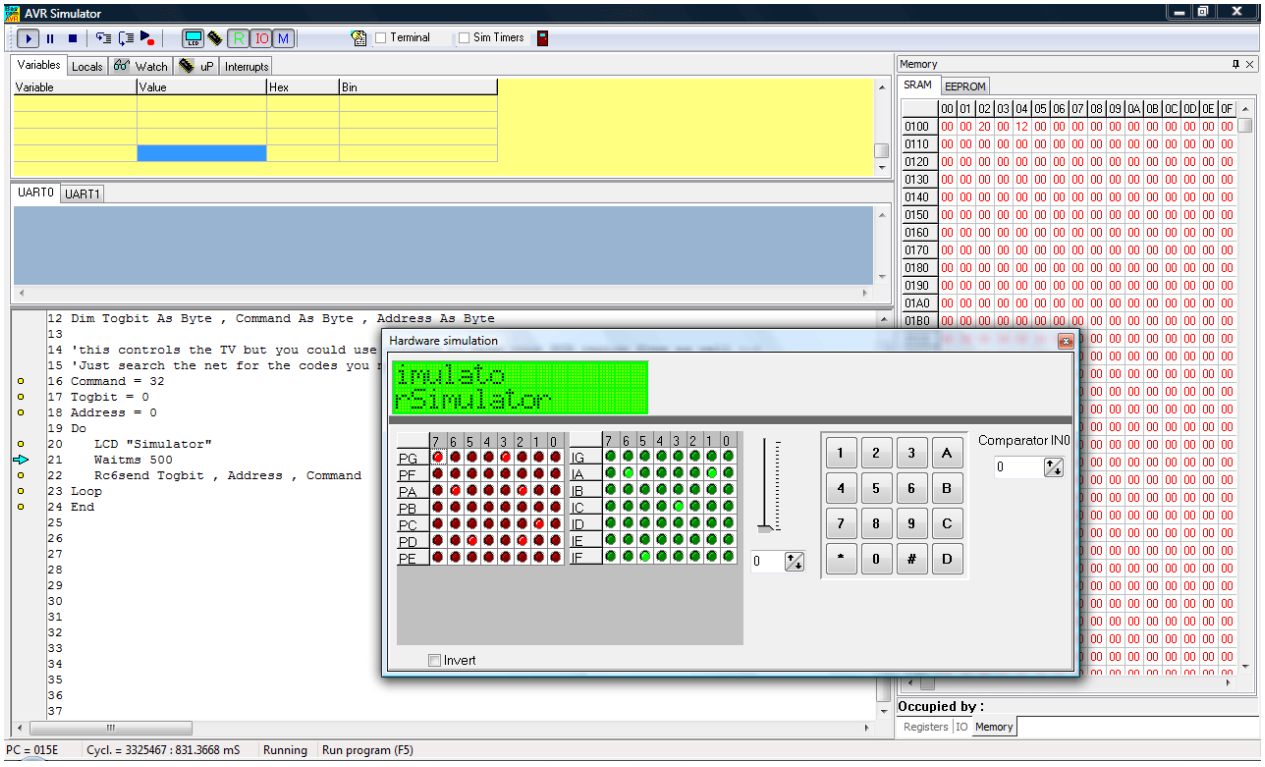
2. واجهة المبرمجة: وفيها يتم برمجة المعالج بعد إجراء عملية توليد الملف البرمجي بالأمر Compile.



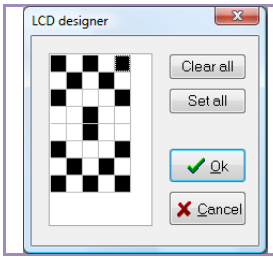
3. واجهة الربط البيني: وفيها يتم عرض المعلومات المرسلة والمتلقاة بين المعالج والحاسب بهدف مراقبة بارامترات النظام بشكل آني.



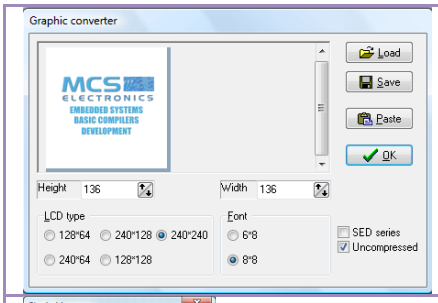
4. واجهة المحاكاة: وفيها يتم تشغيل البرنامج خطوة_خطوة ومراقبة حالة المسجلات الداخلية الذواكر.



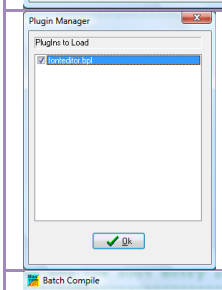
5. بالإضافة إلى الواجهات الأربعة يملك برنامج Bascom-AVR أدوات مساعدة وهي:



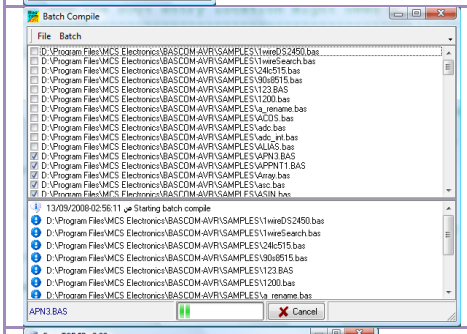
أداة تصميم المحارف (LCD Designer): وتستخدم لتصميم المحارف التي لا توجد على لوحة مفاتيح الحاسب من أجل إظهارها على شاشة الإظهار الكريستالية.



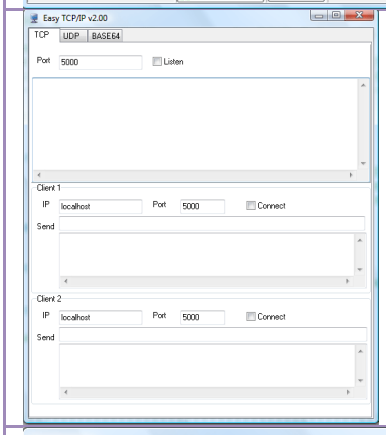
أداة تحويل الصور (Graphic Converter): وتستخدم لتحويل امتداد الصور المراد إظهارها على شاشة الإظهار الرسومية GLCD إلى الصيغة *.bgf



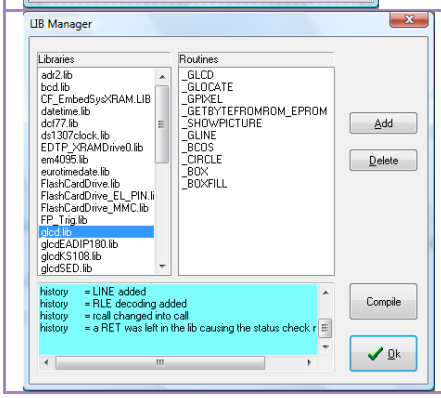
أداة مدير الإضافات (Plugin Manager): وتستخدم لإضافة/حذف الأدوات والموديولات الخارجية.



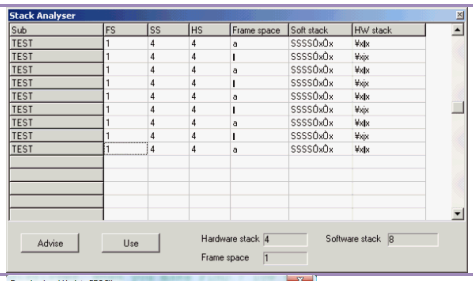
أداة مترجم الملفات المتعددة (Patch Compiler): وتستخدم لتوليد الملف البرمجي لعدة ملفات في آن واحد.



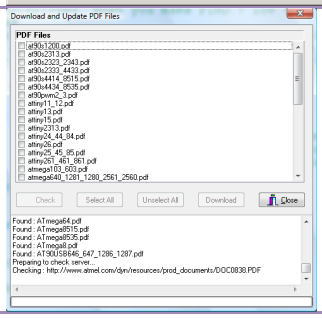
أداة التحكم بالبروتوكول TCP/IP: تستخدم للتحكم ومراقبة المعلومات الموجودة على خط المعطيات.



أدارة إدارة المكتبات (LIB Manager): وتستخدم لإدارة مكتبات البرنامج (حذف \ إضافة).



أداة محلل حالة المكدس (Stack Analyzer): وتستخدم لتحديد حجم المكدس المناسب للتطبيق.



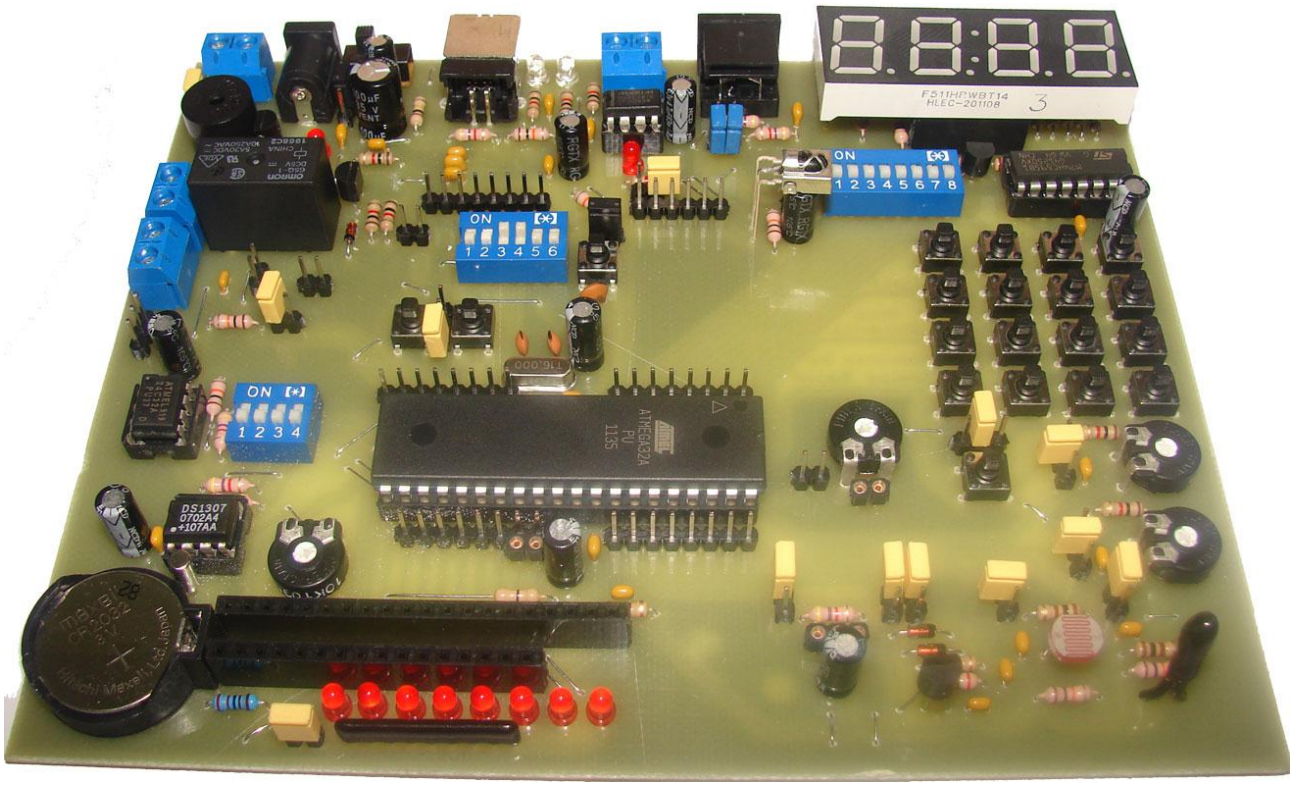
أداة ملفات الوثائق الفنية (PDF Update): تقوم هذه الأداة بالاتصال مع موقع شركة ATMEL وإحضار آخر تحديث للوثائق الفنية للمعالجات المستخدمة من العائلة AVR.

تتميز بيئة التطوير Bascom-AVR بمكتباتها الوظيفية القوية التي تمكن المطور أو المصمم من بناء نظام معقد بزمن قصير نسبياً مقارنة بباقي بيئات التطوير التي تستخدم اللغة C/C++.. الشكل3 يبين تصنيف أنواع التعليمات في البيئة Bascom-AVR حسب وظيفتها.

الشكل3 تصنيف أنواع التعليمات في البيئة Bascom-AVR حسب وظيفتها

11-2 لوحة التطوير المخبرية (The Laboratory Development Board):

لقد تم تصميم هذه اللوحة خصيصاً بحيث تخدم المبتدئ والمتقدم في تعلم برمجة المتحكمات المصغرة من العائلة AVR، حيث تضم أكثر من 25 وحدة محيطية على نفس اللوحة لتغطي ما يقارب 50 تجربة أساسية، وقد تصل إلى أكثر من 75 تجربة بالدمج بين الوظائف المحيطية على اللوحة، بالإضافة إلى إمكانية ربط وحدات خارجية عن طريق وحدات التوسعة المحيطية الموزعة على أطراف اللوحة. يمكن الرجوع إلى الملف "mini-Phoenix-AVR Manual.pdf" من أجل معرفة تفاصيل عن لوحة التطوير (طريقة تجميع اللوحة وميزاتها ومخططاتها التصميمية ...).



الشكل4 لوحة التطوير mini-Phoenix-AVR

12-2 مقدمة إلى متحكمات AVR (Introduction to AVR Microcontrollers):

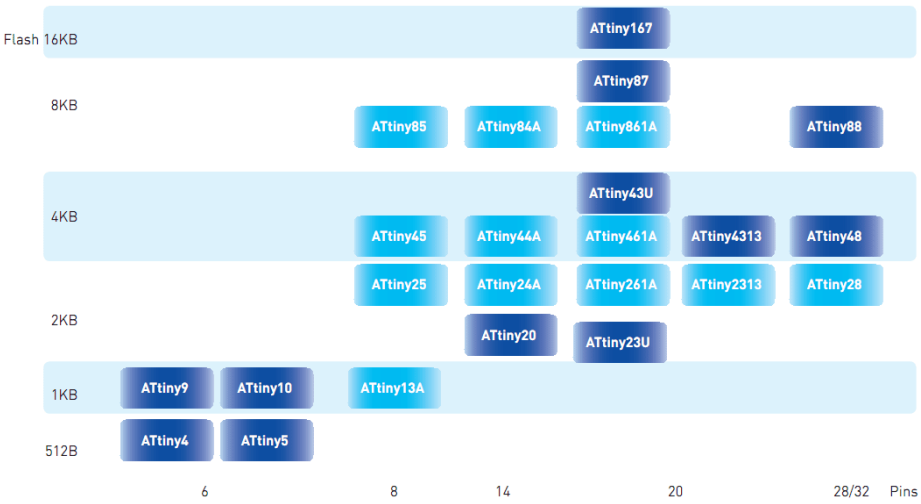
تعتبر متحكمات AVR إحدى منتجات شركة ATMEL الأمريكية، وقد تم تطويرها في مختبرات الشركة الموجودة في النرويج في أواخر التسعينيات، وتعتبر من أكثر المتحكمات المصغرة انتشاراً لما تتميز به من العديد من الميزات التي جعلتها مناسبة لكثير من التطبيقات.

لقد أحدثت شركة ATMEL ثورة في عالم المتحكمات المصغرة بإنتاجها لمتحكمات AVR التي تفوّقت بشكلٍ كبير على العديد من نظيراتها من متحكمات 8-bit، حيث تم استخدام البنية RISC التي تتميز بالأداء العالي وبالطاقة المنخفضة، واحتوت قائمة التعليمات في متحكمات AVR على 132 تعليمة – يُنفذ معظمها خلال دورة آلة واحدة (1-cycle) – وبالتالي عند وصل هزاز 16MHz إلى المتحكم فإنه سينفذ حوالي 16MIPS (مليون تعليمة في الثانية الواحدة)، كما زُوّدت هذه المتحكمات بذاكرة برنامج قابلة للمسح والكتابة لأكثر من 100000 مرة، وضمنت شركة ATMEL أن يبقى البرنامج داخل المتحكم يعمل بشكل صحيح حتى 25 سنة، كما تملك متحكمات AVR وحدات محيطية مدمجة متعددة الوظائف الأمر الذي يوفر استخدام دارات متكاملة خارجية، كذلك زُوّدت معظم متحكمات AVR بمبدل تشابهي رقمي متعدد الأقنية مدمج داخل المتحكم، إضافةً إلى إمكانية برمجة المتحكم دون فصلة عن النظام (In-system Programming)، وكذلك تتوفر في الأسواق بكميات كبيرة وسعرها منخفض مقارنة مع ميزاتها.

13-2 عائلات متحكمات AVR (AVR MCUs Families):

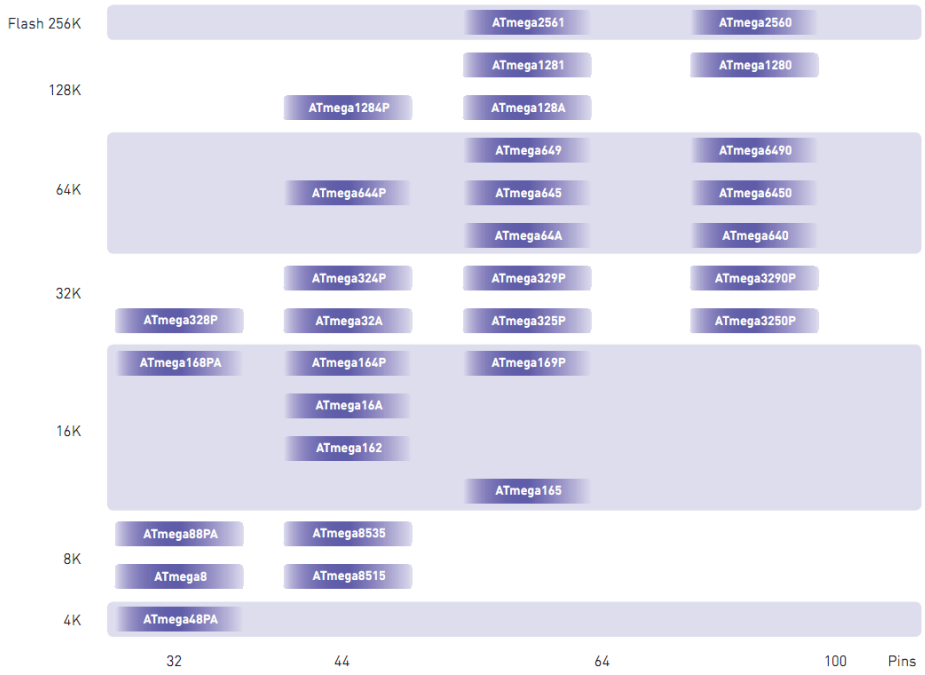
تقسم عائلات متحكمات AVR ذات عرض ناقل 8-bit إلى أربع مجموعات أساسية، إضافة إلى مجموعات أخرى ذات وظائف خاصة، تمتلك جميعها نفس البنية وتختلف عن بعضها البعض بالميزات والخصائص الموجودة في كل نوع:

- ✓ العائلة AT90Sxxxx: العائلة الكلاسيكية التي كان منها الانطلاقة الأولى لمتحكمات AVR في عام 1997 وقد توقف تصنيعها.
- ✓ العائلة ATtinyxx: وهي العائلة الصغرى لمتحكمات AVR المطورة والتي ظهرت في أوائل عام 2000، وهي تملك عدد أقطاب قليل (6~32pin) وحجم ذاكرة برنامج صغير نسبياً (0.5~16KB) وموارد محدودة على الشريحة الأمر الذي يجعل سعرها منخفض مقارنة مع متحكمات ATmega.



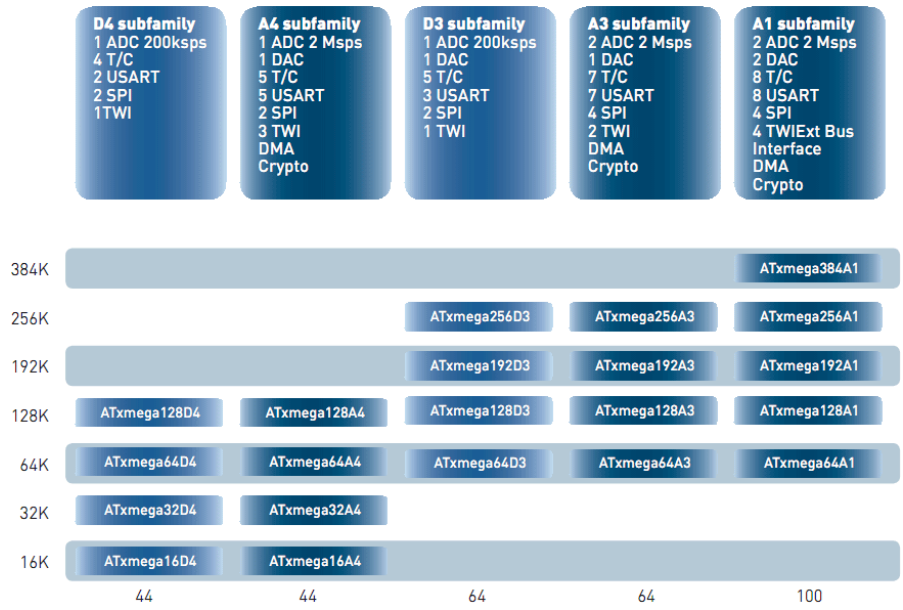
الشكل5 توزع متحكمات العائلة ATinyxxx وفقاً لعدد الأقطاب وحجم ذاكرة البرنامج

✓ العائلة ATmegaxxxx: وهي العائلة الكبرى لمتحكمات AVR المطورة والتي ظهرت في أوائل عام 2000 وهي تملك عدد أقطاب كبير (28~100pin) وحجم ذاكرة برنامج كبير نسبياً (8~256KB) وموارد متنوعة على الشريحة.



الشكل6 توزع متحكمات العائلة ATmega وفقاً لعدد الأقطاب وحجم ذاكرة البرنامج

✓ العائلة ATxmegaxxxx: وهي العائلة المتطورة والأحدث لمتحكمات AVR وقد ظهرت في عام 2008 وهي تملك ميزات متنوعة وسعة معالجة كبيرة نسبياً وتعمل بترددات أعلى من سابقاتها (32MHz)... كما أنها تملك عدد أقطاب كبير (44~100pin) وحجم ذاكرة برنامج كبيرة (16~384KB) إضافة إلى ميزات جديدة لا تتوفر في سابقاتها.



الشكل7 توزع متحكمات العائلة ATxmega وفقاً لعدد الأقطاب وحجم ذاكرة البرنامج

✓ إضافةً إلى العائلات الأربعة الأساسية يوجد عائلات أخرى ذات وظائف وتطبيقات خاصة موضحة في الجدول أدناه.

الصنف	الاستخدام
Automotive AVR	تستخدم في أنظمة التحكم بالمحركات وأنظمة التحكم بالسيارات
Z-Link AVR	تستخدم في بروتوكولات الإرسال الراديوي اللاسلكي في IEEE 802.15.4 / ZigBee
Battery Management AVR	تستخدم للتحكم في شحن المدخرات ومراقبتها وهي تعمل في جهود مرتفعة 1.8~25 فولت
CAN AVR	تستخدم للتحكم بالبروتوكول CAN وتدعم: OSEK، DeviceNet،CANopen
LCD AVR	تستخدم كمعالجات تشغيل أساسية لشاشات الإظهار الكريستالية LCD
Lighting AVR	تستخدم في تطبيقات التحكم الاستطاعية بسرعة المحركات وشدة الإضاءة
USB AVR	تستخدم كموزع أو مخدم للبروتوكول USB

الشكل8 بين ملخصاً للعائلات والخصائص الأساسية لكل منها...

BatteryM AVR	Lighting AVR	USB AVR	megaAVR	tinyAVR
18 ~ 48 Pin	24 ~ 32 Pin	32 ~ 64 Pin	28 ~ 100 Pin	8 ~ 32 Pin
MAX I/O 4~18	MAX I/O 19~27	MAX I/O 22~48	MAX I/O 23~86	MAX I/O 6~28
4KB~40KB Flash	8KB~16KB Flash	8KB~128KB Flash	4KB~256KB Flash	1KB~8KB Flash
256B~1KB EPROM	512B EPROM	512B~4KB EPROM	512B~4KB EPROM	64B~512B EPROM
512B~2KB SRAM	512B~1KB SRAM	512B~8KB SRAM	512B~16KB SRAM	32B~512B SRAM
Up To 8MIPS	Up To 16MIPS	Up To 16MIPS	Up To 20MIPS	Up To 20MIPS
1.8V – 25V	2.7V – 5.5V	2.7V – 5.5V	1.8V – 5.5V	1.8V – 5.5V

Automotive AVR	CAN AVR	LCD AVR	AVR Z-Link	xmegaAVR
14 ~ 64 Pin	64 Pin	64 ~ 100 Pin	MCU Wireless	44 ~ 100 Pin
MAX I/O 6~54	32KB~128KB Flash	MAX I/O 54~69	chipset for:	MAX I/O 36~78
2KB~128KB Flash	1KB~4KB EPROM	16KB~64KB Flash	IEEE 802.15.4	16KB~384KB Flash
128B~4KB EPROM	1K~4KB SRAM	512B~2KB EPROM	and	1KB~4KB EPROM
128B~4KB SRAM	Up To 16MIPS	1KB~4KB SRAM	ZigBee	2KB~32KB SRAM
Up To 16MIPS	2.7V – 5.5V	Up To 20MIPS	applications.	Up To 32MIPS
2.7V – 5.5V		1.8V – 5.5V		1.8V – 3.6V

الشكل8 الخصائص العامة لعائلات متحكمات AVR

14-2 مقارنة بين أشهر عائلات المتحكمات المصغرة (Comparison between most famous µC families):

الجدول التالي يبين مقارنة بين أشهر عائلات متحكمات 8-bit.

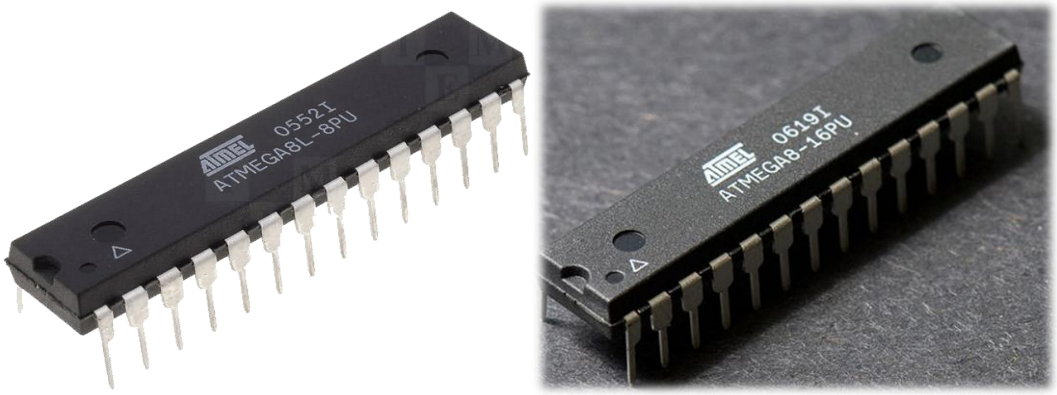
الميزة / العائلة	AVR (Atmel)	8051 (Intel)	PIC (Microchip)	HC11 (Motorola)
البنية الأساسية	Harvard	Von-Neumann	Harvard	Von-Neumann
تقنية النواة	RISC	CISC	RISC	CISC
تردد التشغيل الأعظمي	20MHz	24MHz	20MHz	8MHz
نبضة لكل تعليمة	1	12	4	8
تعليمة في الثانية	16MIPS	2MIPS	5MIPS	1MIPS
عدد التعليمات	132	215	32	200
حجم ذاكرة البرنامج	256KB	32KB	64KB	32KB
عرض ناقل التعليمات	16-bit	8-bit	12-bit	8-bit

من خلال قراءة الجدول نستنتج أفضلية متحكمات AVR للأسباب التالية:

- ✓ متحكمات AVR أسرع من متحكمات PIC بأربع مرات، وأسرع من متحكمات 8051 بثمانية مرات!
- ✓ متحكمات AVR تملك ذاكرة برنامج ذات حجم أكبر من باقي العائلات مما يمكن من كتابة برامج ضخمة.
- ✓ متحكمات AVR مبنية بالاعتماد على تقنية Harvard التي تقوم على الفصل بين ذاكرة البيانات وذاكرة التعليمات بحيث يكون لكل من الذاكرتين خطوط عنونة منفصلة (عناوين فيزيائية مستقلة) وكذلك الأمر بالنسبة لخطوط التحكم وممر المعطيات، الأمر الذي يمكن من أن تحدث عملية قراءة التعليمات مع قراءة أو كتابة البيانات في نفس اللحظة، وكذلك يتيح لطول كلمة البيانات أن يكون مختلفاً عن طول كلمة التعليمات بسبب عدم اشتراك البيانات والتعليمات في نفس الذاكرة. بالمقارنة مع البنية von-Neumann فإن هذه الأخيرة منظمة بحيث لا يوجد فصل بين ذاكرة التعليمات وذاكرة البيانات ولهما نفس خطوط العنونة ونفس ممر المعطيات، بالتالي فإن الفائدتين اللتان تم ذكرهما سابقاً لا توفرهما البنية von Neumann مما يجعل بنية Harvard ذات أداء أعلى من حيث سرعة المعالجة و تنفيذ البرنامج.
- ✓ متحكمات AVR تملك نواة من التقنية RISC التي تمكن من إنجاز تعليمة خلال دورة هزاز واحدة بخلاف التقنية CISC التي تحتاج عدة دورات هزاز لتنفيذ تعليمة واحدة. كذلك فإن البنية RISC أقل تكلفة من البنية CISC.

15-2 قراءة تكويد معالجات العائلة AVR (Reading AVR Package information):

بشكل عام تزود الشرائح المتكاملة والعناصر الإلكترونية برقم أود تكويد خاص بكل صنف. في هذه الفقرة سنشرح تكويد شركة Atmel لمعالجات العائلة AVR بحيث يمكننا معرفة الكثير عن الشريحة من خلال التكويد الموجود على غلافها الخارجي. لنأخذ على سبيل المثال المتحكم المصغر ATmega8 الموضح على الشكل9 نلاحظ أن الغلاف يحوي على رمز الشركة المصنعة (Atmel) ورقم تصنيع تسلسلي (0619I) خاص برقم الدفعة وبيئة عمل المعالج ومن ثم تكويد المعالج وهو: ATmegaa8-16PU.



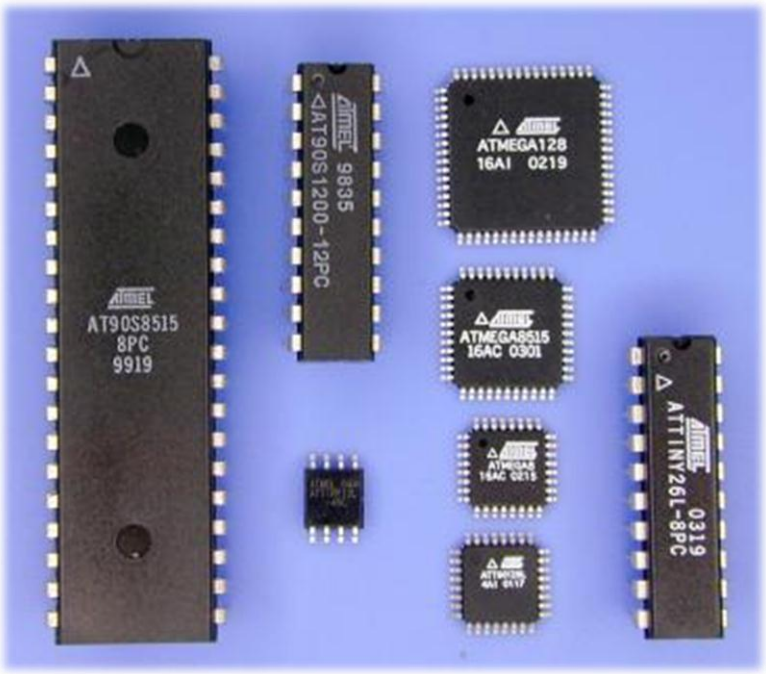
الشكل9 المتحكم المصغر ATmegaa8-16PU

بالنسبة لرمز بيئة عمل المعالج فهو يستخدم للدلالة على نوع التطبيق الذي يمكن أن ستخدم المعالج لأجله، فإما أن يكون تجارياً (**C**) أو صناعياً (**I**) أو عسكرياً (**M**) والاختلاف في ذلك هو من حيث قدرة المعالج على تحمل درجات الحرارة والضجيج العالي:

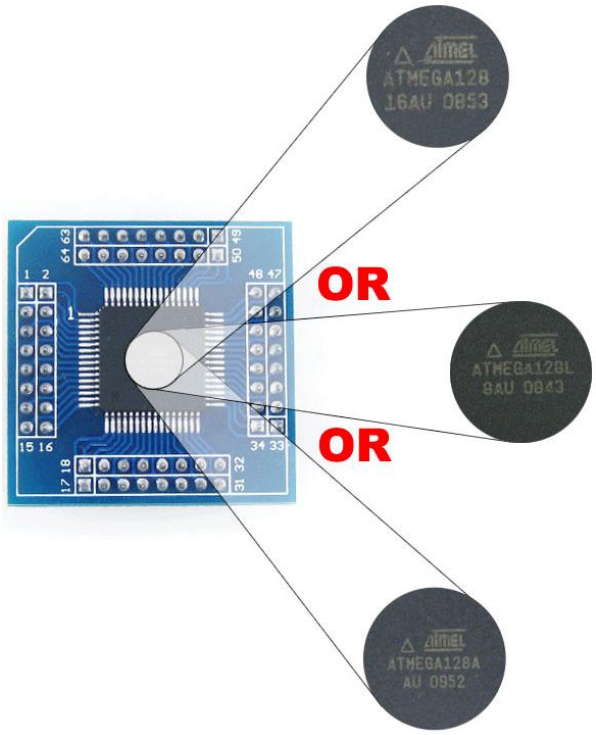
C (Commercial)	**I** (Industrial)	**M** (Martial)
تطبيقات تجارية	تطبيقات صناعية	تطبيقات عسكرية

بالنسبة لكود المعالج فهو على الشكل التالي:

الرمز	الدلالة
AT	اختصار لاسم الشركة المصنعة ATMEL.
Mega	العائلة التي ينتمي إليها هذا المعالج (Xmega، Mega، Tiny،90S).
8	هذا الرقم يعبر عن حجم ذاكرة البرنامج كيلو بايت (8، 16، 32، 64، 128، 256 KB).
L	في حال أن الكود يملك حرف L فهذا يعني أنه قابل على العمل بجهود منخفضة (2.7V~5.5V) وبدون هذا الحرف فهذا يعني أن المعالج يعمل عند جهود (4.5V~5.5V).
8	تردد العمل الأعظمي للمعالج (8، 16، 20، 24، 36).
PU	شكل غلاف الشريحة، فإما أن تكون من نوع يتم لحامه على الطبقة السفلية للدارة المطبوعة ورمزه PU (PDIP)، أو من النوع السطحي الذي يتم لحامه على الطبقة العلوية للدارة المطبوعة ورمزه AU (TQFP).



الشكل10 نموذج من متحكمات AVR بغلاف خارجي من النوع PDIP والنوع TQFP



من الجدير ذكره أن شركة Atmel قامت بدمج المعالج ذو اللاحقة ATmegaxxxL-8xx (يعمل عند جهد منخفض 2.7V وتردد منخفض 8MHz) مع المعالج ATmegaxxx-16xx (يعمل عند جهد 4.5V وتردد حتى 16MHz) في معالج جديد متطور ATmegaxxxA-xx يجمع خصائص كلا المعالجين؛ هذا الأخير يعمل عند جهود من 2V5 – 5V5 وتردد 1MHz – 16MHz وقد صدر هذا الإصدار من المتحكمات في عام 2010 ليحل محل الإصدارات القديمة ذات اللاحقة L-8 واللاحقة 16-.

الشكل جانباً يبين مقارنة بين الحالات الثلاث لتكويد المتحكم ATmega128 حيث:

ATmega128L-8AU (قديم لم يعد في طور التصنيع).

ATmega128-16AU (قديم لم يعد في طور التصنيع).

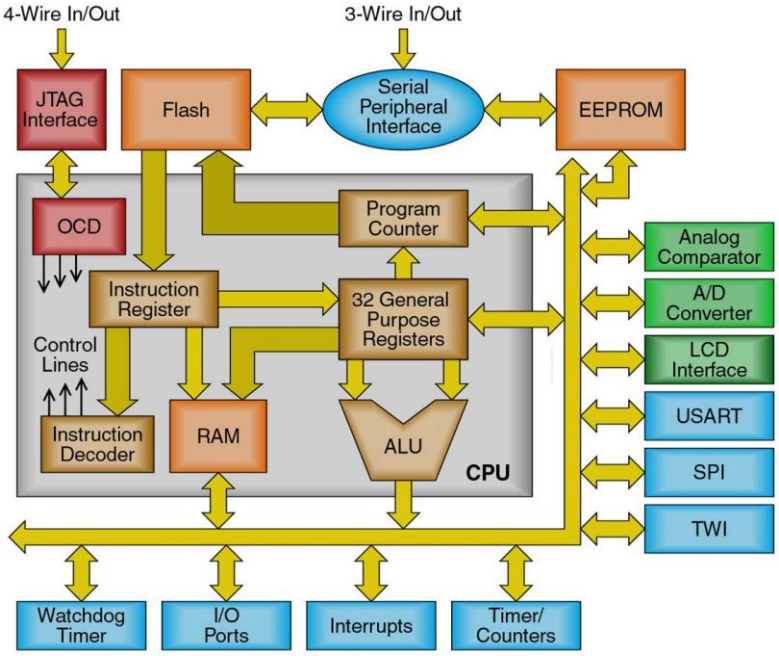
ATmega128A-AU (الجيل الجديد وهو موافق تماماً لسابقيه).

16-2 نواة معالجات العائلة AVR (AVR CPU Core):

تتمثل المهمة الرئيسية لنواة المعالج في ضمان تنفيذ البرنامج بشكل صحيح والذي بدوره يتطلب توفير إمكانية الولوج إلى الذواكر والقدرة على تنفيذ العمليات الحسابية والتحكم بالطرفيات... إضافة إلى التعامل مع المقاطعات المختلفة.

2-17 نظرة عامة على البنية الداخلية للنواة (CPU Core Architectural Overview):

تعتمد العائلة AVR البنية Harvard بهدف زيادة مستوى الأداء والعمل المتوازي – بحيث تكون هناك ذاكرة مخصصة للبيانات مع ممر معطيات خاص وذاكرة أخرى منفصلة لتعليمات البرنامج مع ممر خاص أيضاً. يتم تنفيذ التعليمات في ذاكرة البرنامج بمستوى واحد من المعالجة التفرعية بحيث أنه في الوقت الذي تقوم فيه وحدة المعالجة المركزية (CPU) بتنفيذ إحدى التعليمات يتم إحضار شفرة التعليمة التالية لها من ذاكرة البرنامج؛ إن هذا المبدأ يسمح بتنفيذ تعليمة عند كل نبضة ساعة.



الشكل 11 مخطط البنية الداخلية لنواة متحكمات AVR

يرتبط مع وحدة الحساب والمنطق ALU (Arithmetic Logic Unit) ملف المسجل ذو الولوج السريع والذي يحتوي على 32 مسجلاً للأغراض العامة كل منها بطول 8-bit وبزمن ولوج قدره نبضة واحدة من نبضات الهزاز – ويقصد بالولوج السريع للملف المسجلات على أنه الولوج الذي يستغرق زمن قدره دورة ساعة واحدة؛ هذا يعني أنه خلال دورة ساعة واحدة تقوم وحدة الحساب والمنطق بتنفيذ عملية واحدة، فهي تقوم أولاً بإخراج المعاملين من ملف السجلات، ومن ثم تنفذ العملية، ومن ثم تعيد تخزين النتيجة في ملف المسجلات، وكل ذلك يتم خلال دورة ساعة واحدة.

2-18 وحدة الحساب والمنطق (Arithmetic Logic Unit):

تدعم وحدة الحساب والمنطق العمليات الحسابية والمنطقية بين محتوى المسجلات أو بين محتوى المسجل وقيمة ثابتة، كما يمكن إجراء عملية معينة على محتوى مسجل وحيد، وبعد تنفيذ العملية يتم تحديث مسجل الحالة ليعطي المعلومات عن ناتج تلك العملية. تتمتع وحدة الحساب والمنطق ALU عالية الأداء في العائلة AVR بالارتباط المباشر مع مسجلات العمل الاثنين والثلاثين ذات الأغراض العامة، فخلال دورة ساعة واحدة تنفذ وحدة الحساب والمنطق عملية ما بين مسجلين في ملف المسجلات أو بين مسجل وقيمة ثابتة.

ويمكن تقسيم عمليات ALU إلى ثلاث فئات رئيسية: فئة حسابية – فئة منطقية – فئة العمليات على مستوى البت (Bit). كما تملك معالجات العائلة AVR في بنيتها ضارب متطور يدعم عمليات ضرب للأعداد المؤشرة وغير المؤشرة والاعداد الكسرية.

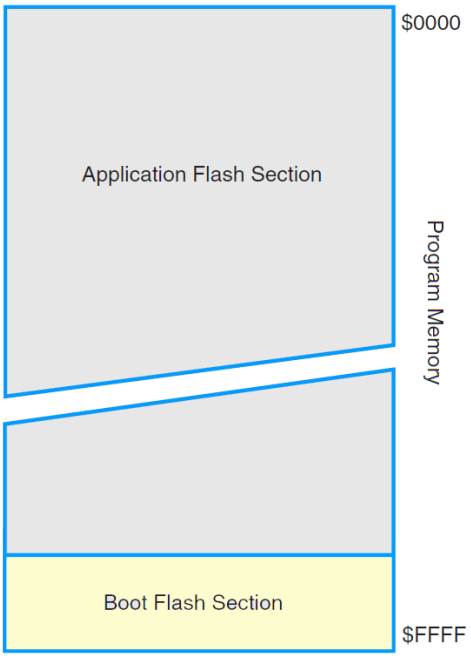
19-2 الذواكر في متحكمات AVR (Memories in AVR MCUs):

تمتلك معظم متحكمات AVR ثلاث أنواع من الذواكر المدمجة على الشريحة وهي: ذاكرة البيانات (Data Memory) – ذاكرة البرنامج (Flash Memory) – ذاكرة المعطيات (EEPROM).

19-2-1 ذاكرة البرنامج (Flash Memory):

يملك المتحكم ATmega32A ذاكرة وميضية من النوع Flash مدمجة ضمن الشريحة بحجم 32KB ($7FFF - $0000) مخصصة لتخزين برنامج عمل المتحكم المصغر، وبما أن جميع تعليمات متحكمات AVR هي ذات شفرة وحيدة بطول كلمة واحدة (16-bit) أو كلمتين (32-bit)، فقد تم تنظيم ذاكرة البرنامج لتشكل 16K x 16، كما تم تقسيمها إلى قسمين منفصلين: قسم برنامج الإقلاع (Boot Program) وقسم برنامج التطبيق (Application Program) وذلك بهدف الحماية من سرقة البرنامج.

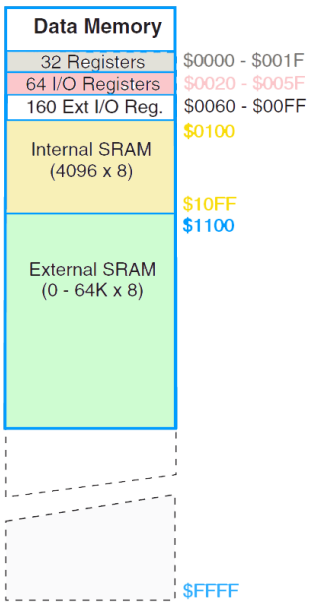
تتميز ذاكرة البرنامج في متحكمات AVR بديمومة 10,000 دورة مسح/كتابة على الأقل، كما أنها قابلة لإعادة البرمجة في النظام دون الحاجة إلى نقل المتحكم إلى مبرمجة خاصة وهو ما يدعى بـ"In System Programing". كذلك يملك المتحكم ATmega32A عداد برنامج (PC) بطول 16-bit (وهو متغير الطول بالنسبة بكل معالج) وبذلك يستطيع عنونة كامل مجال ذاكرة البرنامج الذي هو بطول 16KB x 16bit.



الشكل12 خريطة ذاكرة البرنامج

19-2-2 ذاكرة البيانات SRAM (SRAM Data Memory):

وهي عبارة عن ذاكرة وصول عشوائي (Static Random Access Memory) مؤقتة يتم فيها إجراء العمليات على المتحولات؛ يملك المتحكم ATmega32A ذاكرة بيانات من النوع SRAM بطول 2KB.



الشكل13 تنظيم الذاكرة الدخلية SRAM للمتحكم ATmega128A

19-2-3 ذاكرة المعطيات EEPROM (EEPROM Data Memory):

يملك المتحكم ATmega32A ذاكرة معطيات من النوع EEPROM بطول 1KB وهي منظمة مستقل لتخزين البيانات، ويمكن فيه القراءة من أو الكتابة على أي بايت من بايتات هذه الذاكرة بشكل مستقل، كما تسمح ديمومة الذاكرة EEPROM بأكثر من 100,000 عملية كتابة ومسح.

20-2 مصادر التوقيت في متحكمات AVR (Clock Sources in AVR MCUs):

تملك متحكمات AVR العديد من مصادر إشارة التوقيت والتي تقسم بشكل رئيسي إلى خمسة مصادر:

- هزاز كريستالي/سيراميكي خارجي (External Crystal/Ceramic Resonator).
- هزاز كريستالي خارجي ذو تردد منخفض (External Low-frequency Crystal).
- هزاز RC خارجي (External RC Oscillator).
- هزاز RC داخلي معاير (Calibrated Internal RC Oscillator).
- إشارة توقيت خارجية (External Clock).

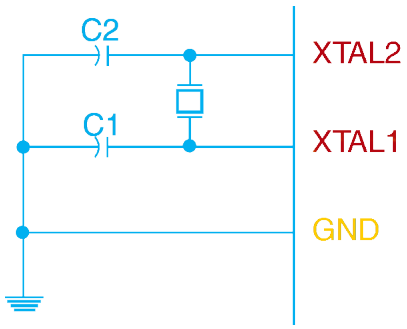
يتم ضبط مصدر إشارة التوقيت من خلال ضبط الفيوزات الداخلية للمتحكم والمسؤولة عن مصدر التوقيت (CKSEL3:0). لمزيد من التفاصيل يمكن الرجوع إلى الوثيقة الفنية للمتحكم/الفقرة "8.2 - Clock Sources".

20-2-1 الهزاز RC الداخلي للمتحكم (Calibrated Internal RC Oscillator):

تملك معظم متحكمات AVR وخصوصاً العائلة ATmega هزاز RC داخلي معاير مخبرياً بتردد ثابت يمكن ضبطه من أجل القيم 1MHz, 2MHz, 4MHz, 8MHz، وإن الحالة الافتراضية لمصدر إشارة التوقيت لمعظم متحكمات AVR-Mega هو "هزاز RC داخلي معاير" بتردد 1MHz ويمكن تغييره من خلال الفيوزات الداخلية للمتحكم الخاصة بمصدر إشارة التوقيت (CKSEL3:0).

20-2-2 وصل هزاز كريستالي خارجي (External Crystal Oscillator):

من أجل دقة أعلى لإشارة التوقيت، أو من أجل تشغيل المعالج عند تردد عمل أكبر من 8MHz، فإنه يمكن استخدام هزاز كريستالي خارجي يتم وصله مع المتحكم عبر القطبين XTAL1 (الدخل) والقطب XTAL2 (الخرج) مع مراعات إضافة مكثفات تحميل خارجية بقيمة 12-22pF كما هو موضح على الشكل14. من الضروري إعادة ضبط الفيوزات الداخلية للمتحكم والخاصة بمصدر إشارة التوقيت (CKSEL3:0) على "External Crystal Resonator".



الشكل14 وصل هزاز كريستالي خارجي مع المتحكم المصغر

20-2-3 وصل هزاز كريستالي خارجي ذو تردد منخفض (External Low-frequency Crystal):

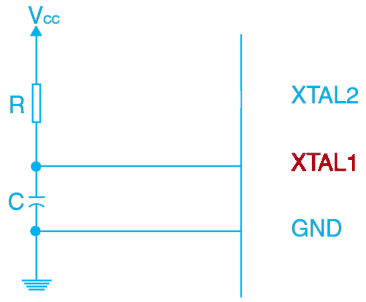
يمكن وصل هزاز كريستالي خارجي بتردد منخفض 32.768KHz كمصدر توقيت مخصص لتطبيقات الساعات الرقمية بعد إعادة تعيين الفيوزات الداخلية للمتحكم الخاصة بمصدر إشارة التوقيت (CKSEL3:0) على "External Low-frequency Crystal" حيث CKSEL = "1001".

20-2-4 وصل هزاز RC خارجي (External RC Oscillator):

من أجل التطبيقات التي لا تحتاج إلى دقة في إشارة التوقيت، فإنه يمكن استخدام دارة RC لتوليد إشارة التوقيت (100Hz~12MHz) كما هو مبين على الشكل15. القيمة التقريبية للتردد المولد يمكن حسابها من المعادلة:

$$f = \frac{1}{3 \times R \times C}$$

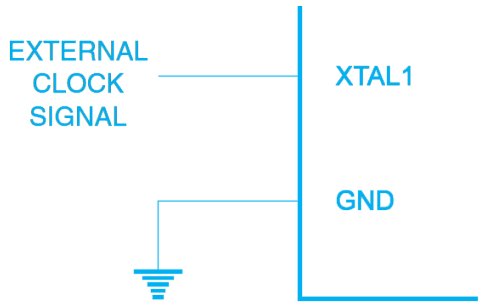
حيث أن قيمة C يجب أن تكون أكبر من 22pF. كما أن القيمة الأعظمية للتردد f يجب أن لا تتجاوز 12MHz. يجب ضبط الفيوزات الداخلية للمتحكم الخاصة بمصدر إشارة التوقيت (CKSEL3:0) على "External RC Oscillator".



الشكل15 وصل هزاز RC خارجي مع المتحكم المصغر

20-2-5 إشارة توقيت خارجية (External Clock):

يمكن كتطبيق إشارة توقيت خارجية مباشرة على القطب XTAL1 بعد إعادة تعيين الفيوزات الداخلية للمتحكم الخاصة بمصدر إشارة التوقيت (CKSEL3:0) على"External Clock" حيث "0000" = CKSEL. الشكل16 يبين طريقة تطبيق الإشارة.



الشكل16 تطبيق إشارة توقيت خارجية للمتحكم

2-21 الدراسة التطبيقية – المتحكم المصغر ATmega32A (Case Study – Atmega32A):

السبب في اختيار المعالج ATmega32 هو أنه يعتبر من المعالجات المتقدمة في العائلة ATmegaxxx ويضم معظم الميزات والوحدات المحيطية المتوفرة في العائلة Mega وبالتالي فإننا ندرس الآن الحالة الأعم والأشمل، كما أنه متوفر بغلاف من النوع PDIP.

بشكل عام وعند شراء دارة متكاملة ما، فإن أول ما يحتاج إليه هو معرفة خصائصها وميزاتها وكيفية عملها وتوزع أقطابها، وهذا يتم من خلال قراءة المعلومات المهمة من الوثيقة الفنية (Datasheet) الخاصة بالدارة المتكاملة. وهنا أود التنويه إلى أنه عند استخدام لغات البرمجة عالية المستوى فإن المبرمج لن يحتاج إلى قراءة الوثيقة الفنية للمتحكم المصغر كاملةً من أجل برمجته، وهذا بدوره يختصر وقتاً كبيراً في تعلم برمجة المتحكمات المصغرة بدون اللجوء إلى دراسة البنية الداخلية للمعالج مفصلةً كما هو الحال عند البرمجة بلغة التجميع (Assembly). لذلك، سوف أشرح المعلومات التي تفيدنا في الوثيقة الفنية كالميزات الأساسية للمعالج وتوزع الأقطاب، مع العلم أنه لابد – لاحقاً – من العودة إلى بعض التفاصيل في البنية الداخلية للمعالج عن مرحلة متقدمة.

2-22 الميزات الأساسية للمتحكم ATmega32A (ATmega128A Features):

الميزات الأساسية تأتي دائماً في الصفحة الأولى من الوثيقة الفنية لأي دارة متكاملة... في ما يلي ميزات المعالج ATmega32A.

- ✓ متحكم 8-bit بأداء عالٍ واستهلاك منخفض للطاقة.
- ✓ بنية متطورة من النوع RISC (أقل عدد ممكن من التعليمات):
  - 133 تعليمة معظمها تنفذ بدورة آلة واحدة
  - 32 x 8 مسجلات أغراض عامة + ومسجلات تحكم محيطية
  - عمل مستقر ومناعة ضد الضجيج
  - قادر على تنفيذ 16 مليون تعليمة في الثانية عند تردد 16MHz
  - يحوي على مضاعف دورة العمل
- ✓ ذاكرة معطيات دائمة:
  - 32KB ذاكرة برنامج يمكن برمجتها بدون فصل المعالج عن الدارة، قابلة للمسح والكتابة 10000 مرة.
  - أقفال برمجية مستقلة مع قطاع مخصص لكود إقلاع.
  - 1KB ذاكرة معطيات دائمة EEPROM قابلة للمسح والكتابة 100000 مرة.
  - 2KB ذاكرة وصول عشوائي مؤقتة SRAM
  - إمكانية عنونة 64KB (وصل) ذاكرة برنامج خارجية
  - أقفال برمجية من أجل حماية البرنامج على الشريحة
  - واجهة ربط تسلسلية (SPI) من أجل برمجة المعالج دون فصله
- ✓ واجهة اختبار (JTAG):

- قابلية مسح المسجلات الداخلية للمعالج وقراءة حالاتها
- دعم متقصي أخطاء (Debug) شامل للشريحة
- إمكانية برمجة ذاكرة البرنامج وذاكرة المعطيات.

✓ الميزات المحيطية:

- مؤقت/عداد 8-bit عدد 2 مزود بأنماط مقسم ترددي وحادثة مقارنة
- مؤقت/عداد 16-bit موسع مزود بأنماط مقسم ترددي وحادثة مقارنة وحادثة مسك
- عداد الزمن الحقيقي مع هزاز مستقل
- أربعة قنوات خرج (PWM) تعديل عرض النبضة 16-bit مع إمكانية التحكم بالدقة من 2 وحتى 16 بت
- ثمان قنوات تبديل تشابهي/رقمي بدقة 10-bit
- قناتي مداخل تفاضلية للتبديل تشابهي/رقمي بدقة 10-bit مع دارة ربح 1x, 10x, or 200x
- نافذة اتصال تسلسلية ثنائية (I2C)
- نافذتي اتصال تسلسلي (USARTs) قابلة للبرمجة
- نافذة اتصال تسلسلية (SPI) بنمطي عمل قائد/تابع
- مؤقت مراقبة قابل للبرمجة مع هزاز مستقل
- نافذة مقارن تشابهي

✓ الميزات الخاصة للمعالج:

- تصفير عند وصل التغذية وكاشف انخفاض جهد التغذية للشريحة
- هزاز داخلي معاير
- مصادر مقاطعة خارجية وداخلية
- ستة أنماط لتخفيض الطاقة ولتخفيض ضجيج المبدل
- إمكانية تحديد تردد الهزاز الداخلي برمجياً
- إلغاء شامل لمقاومات الرفع الداخلية للبوابات

✓ عدد أقطاب الدخل/الخرج وشكل الغلاف الخارجي المعالج:

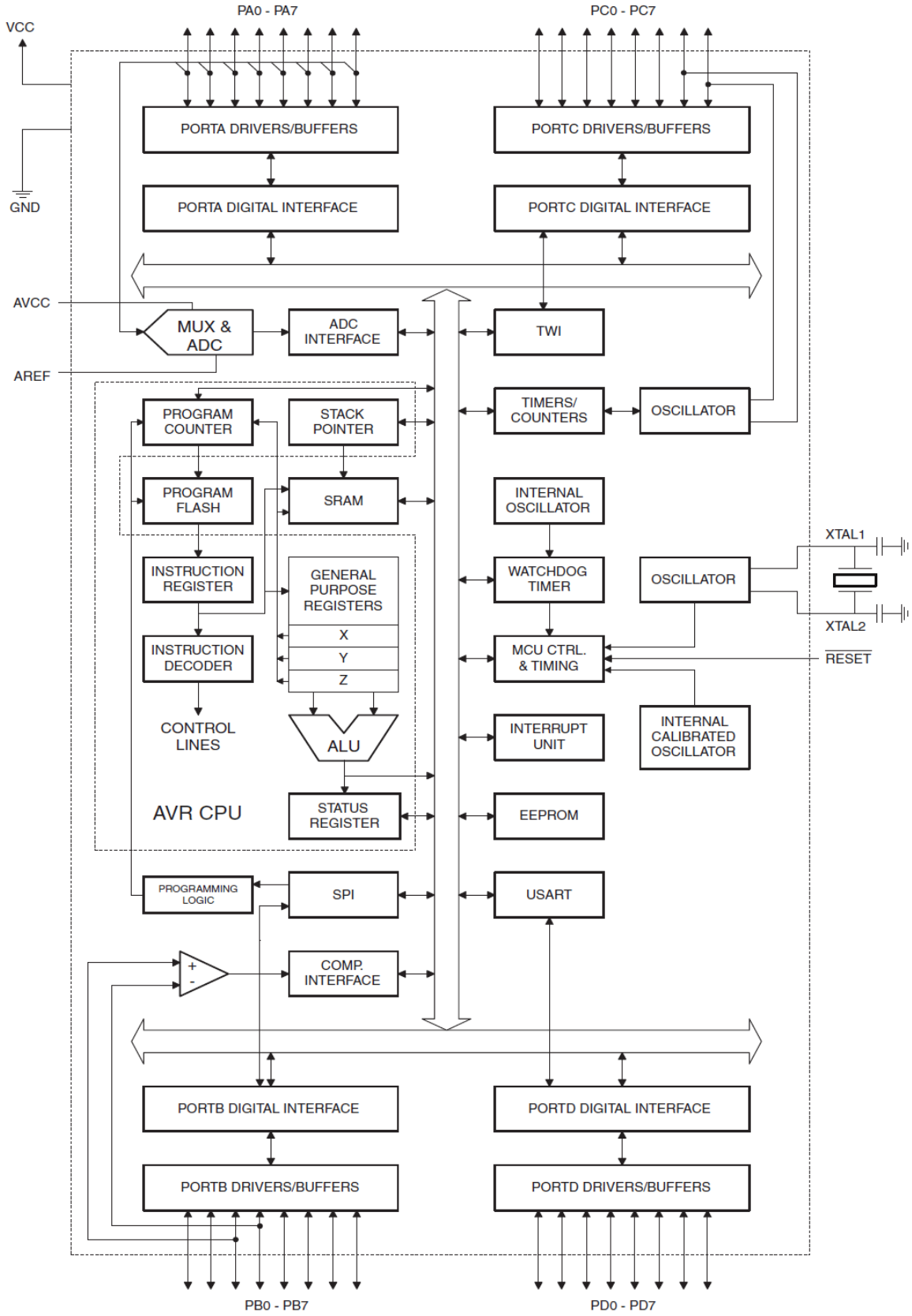
- 32 قطب دخل/خرج قابل للبرمجة متوفر من أجل شريحة 40 قطب بغلاف PDIP.

✓ جهود العمل للشريحة في المجال 2.7 - 5.5V

✓ تردد عمل أعظمي حتى 0 - 16MHz

✓ استهلاك الطاقة في النمط الفعال (Active) 0.6mA وفي نمط البطالة (Idle) 0.2mA وفي نمط الطاقة التحتية 1uA.

23-2 مخطط البنية الداخلية للمتحكم ATmega32A (ATmega32A Block Diagram):



الشكل17 المخطط الصندوقي للبنية الداخلية للمتحكم ATmega32A

لقد تم صناعة الشريحة ATmega32A باستخدام تقنية ذواكر ATMEL الغير قابلة للزوال ذات الكثافة العالية، مع إمكانية برمجة ذاكرة البرنامج الوميضية (Flash) المبنية على شريحة المتحكم إما من خلال الوصلة التسلسلية SPI أو باستخدام مبرمجة تفرعية أو باستخدام برنامج إقلاع موجود على الشريحة (Boot program) حيث تستطيع البرمجية المخزنة في جزء الإقلاع (Bootloader) في الذاكرة الوميضية متابعة عملها أثناء تحديث القسم الرئيسي في ذاكرة البرنامج. لقد أدى الجمع ما بين معالجات RISC ذات 8-bit مع ذاكرة البرنامج القابلة لإعادة البرمجة إلى إنتاج المتحكم ATmega132A الذي يتمتع بالقوة و المرونة العالية وبالكلفة المنخفضة للعديد من تطبيقات التحكم المتطورة. الشكل17 يبين المخطط الصندوقي للبنية الداخلية للمتحكم ATmega32 وهو يبين طريقة ربط الوحدات المحيطية والمسجلات مع وحدة المعالجة المركزية.

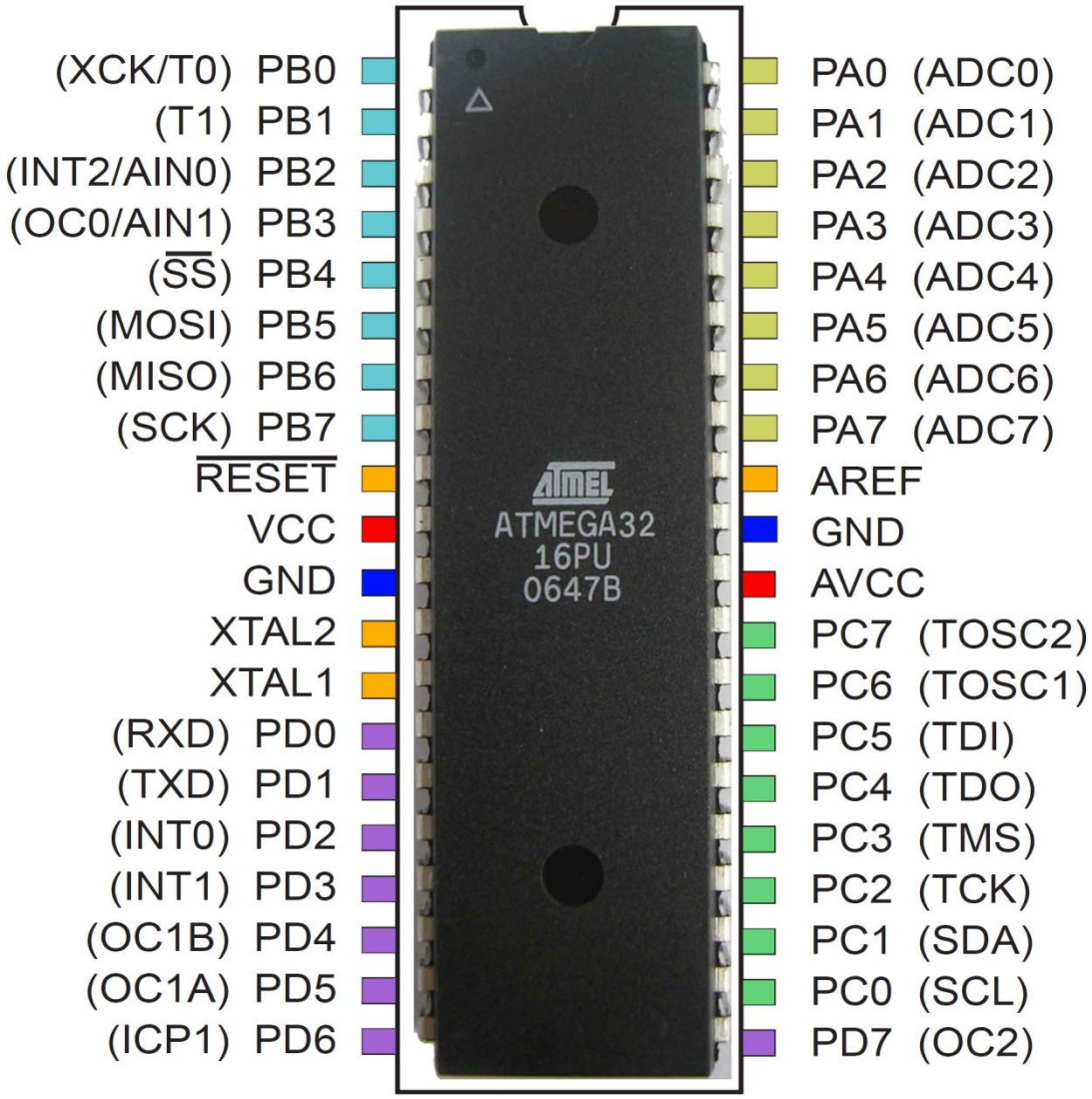
24-2 تصنيفات الأقطاب وظيفياً في متحكمات AVR (AVR MCUs Pins Functional Classification):

يمكن تصنيف أقطاب متحكمات العائلة AVR من حيث وظائفها على الشكل التالي:

1. أقطاب التغذية الرقمية (VCC, GND).
2. أقطاب التغذية التشابهية (AVCC, AGND).
3. أقطاب الدخل والخرج (Input / Output Pins = Ports).
4. أقطاب البرمجة:
   - الواجهة البرمجية ISP (MISO, MOSI, SCK, RESET).
   - الواجهة البرمجية PDI (PDO, PDI).
5. أقطاب المقاطعات الخارجية (INT0 – INT7).
6. أقطاب مصادر التوقيت (XTAL1, XTAL2, TOSC1, TOSC2).
7. أقطاب المؤقتات/العدادات (T0, T1, T2, ICP).
8. أقطاب المبدلات التشابهية الرقمية (ADC0 – ADC7).
9. أقطاب النوافذ التسلسلية:
   - النافذة التسلسلية UART / USART (XCK, TXD, RXD).
   - النافذة التسلسلية SPI (MISO, MOSI, SCK, SS).
   - النافذة التسلسلية I2C (SDA, SCL).
10. أقطاب المقارن التشابهي (AIN0, AIN1).
11. أقطاب إشارات PWM (OCA1, OCB1, OCC1, …).
12. أقطاب نافذة المراقبة JTAG (TDI, TDO, TMS, TCK).
13. أقطاب الوصل مع ذاكرة خارجية XRAM (AD0-7, A8-15, WR/RD).

25-2 وصف أقطاب المتحكم ATmega32A (ATmega32A Pin Description):

يملك المتحكم ATmega32A مجموعة من الأقطاب عددها 40 قطب موزعة على الأطراف الفيزيائية لشريحة المتحكم وهي:



الشكل18 توزع الأقطاب على الشريحة ATmega32A

الاسم	الوظيفة
VCC	قطب جهد التغذية الموجب VCC = 2.5 – 5.5V
GND	قطب جهد التغذية الصفري (الأرضي) GND = 0V
Port A (PA7:PA0)	البوابة A: وهي عبارة عن بوابة دخل/خرج ذات ثمانية أقطاب ثنائية الاتجاه، وقد زودت الأقطاب بمقاومات رفع داخلية (Pull-up) مع إمكانية اختيار مقاومة الرفع لكل قطب على حدى. تمتلك البوابة A وظيفة ثانوية أخرى وهي قنوات المبدلات التشابهية الرقمية (ADC0-ADC7).

Port B (PB7:PB0)	البوابة B: وهي عبارة عن بوابة دخل/خرج ذات ثمانية أقطاب ثنائية الاتجاه، وقد زودت الأقطاب بمقاومات رفع داخلية (Pull-up) مع إمكانية اختيار مقاومة الرفع لكل قطب على حدى، كذلك تمتلك البوابة B وظائف ثانوية أخرى وهي: واجهة اتصال تسلسلية SPI (SS, SCK, MOSI, MISO)، وقطب توليد إشارة PWM (OC0)، وأقطاب المقارن التشابهي (AIN0, AIN1)، المقاطعة الخارجية (INT2)، العدادات (T0/T1).
Port C (PC7:PC0)	البوابة C: وهي عبارة عن بوابة دخل/خرج ذات ثمانية أقطاب ثنائية الاتجاه، وقد زودت الأقطاب بمقاومات رفع داخلية (Pull-up) مع إمكانية اختيار مقاومة الرفع لكل قطب على حدى. تمتلك البوابة C وظائف ثانوية أخرى حيث تعمل كنافذة تتبع أخطاء JTAG (TCK, TMS, TDO, TDI)، نافذة تسلسلية I2C (SCK, SDA).
Port D (PD7:PD0)	البوابة D: وهي عبارة عن بوابة دخل/خرج ذات ثمانية أقطاب ثنائية الاتجاه، وقد زودت الأقطاب بمقاومات رفع داخلية (Pull-up) مع إمكانية اختيار مقاومة الرفع لكل قطب على حدى. تمتلك البوابة D وظائف ثانوية أخرى وهي: المقاطعات الخارجية (INT0-INT1)، النافذة التسلسلية USART (TXD, RXD) ، مدخل حادثة المسك للموقت1 (ICP1). وأقطاب توليد إشارات PWM (OC2, OC1B, OC1A).
RESET	مدخل تصفير الشريحة؛ عند تطبيق إشارة كهربائية ذات منطق منخفض على القطب RESET لمدة دورتي آلة، فإن دارة التصفير الداخلية تعمل على تصفير المتحكم – عداد البرنامج PC = 0.
XTAL1	مدخل الهزاز الخارجي: وهو عبارة عن مدخل دارة مضخم الهزاز العاكس.
XTAL2	مدخل الهزاز الخارجي: وهو عبارة عن خرج دارةمضخم الهزاز العاكس.
AVCC	قطب التغذية للمبدل التشابهي الرقمي (ADC)، إذا كان المبدل ADC غير مُستخدم فإن هذا القطب يجب وصله إلى القطب Vcc، أما إذا كان المبدل ADC مُستخدماً فيوصل هذا القطب مع Vcc عن طريق مرشح تمرير مُنخفض.
AGND	إذا كانت الدارة التي نقوم بتصميمها لها أرضي تشابهي مستقل، فيجب ربط هذا القطب مع هذا الأخير، وإلا يُربط هذا القطب مع القطب الأرضي العام GND.
AREF	مدخل الجهد المرجعي التشابهي للمبدل ADC ويجب أن تتراوح قيمته عند عمل المبدل ما بين A_{Vcc} – 2V.

. . . ﴿انتهت الجلسة العملية الثانية﴾ . . .

وليد بليد

– دمتم بخير ومودة ونور –