

AVR e Arduino

Técnicas de Projeto

Charles Borges de Lima

e

Marco V. M. Villaça

AVR e Arduino Técnicas de Projeto

2ª edição

Florianópolis

Edição dos Autores

2012

Copyright © 2012 para Charles Borges de Lima e Marco Valério Miorim Villaça.

Todos os direitos reservados e protegidos pela Lei 9.610 de 19/02/1998. É proibida a reprodução desta obra, mesmo parcial, por qualquer processo, sem prévia autorização, por escrito, dos autores.

Figura da capa de www.dreamstime.com.

L732a Lima, Charles Borges de
 AVR e Arduino : técnicas de projeto / Charles Borges de Lima,
 Marco Valério Miorim Villaça. 2. ed. – Florianópolis: Ed. dos autores,
 2012.
 632 p.: il; 21,0 cm.

 Inclui bibliografia
 ISBN: 978-85-911400-1-5

 1. Microcontroladores – AVR. 2. Arduino. 3. Engenharia eletrônica.
 4. Projetos eletrônicos. 5. Programação. I Villaça, Marco Valério
 Miorim. II. Título.

CDD:

621.381

Sistema de Bibliotecas Integradas do IFSC.

Biblioteca Dr. Hercílio Luz – Campus Florianópolis.

Catalogado por: Rose Mari Lobo Goulart, CRB 14/277.

Todas as marcas registradas, nomes ou direitos de uso citados neste livro pertencem aos seus respectivos proprietários.

As técnicas de projetos eletrônicos apresentadas não são exclusivas, são técnicas comuns empregadas com microcontroladores. Elas estão descritas em referências técnicas, sendo informações empregadas no meio industrial e acadêmico.

Os autores acreditam que todas as informações aqui apresentadas estão corretas e podem ser utilizadas para qualquer fim legal. Entretanto, não existe qualquer garantia implícita ou explícita, de que o uso de tais informações conduzirá sempre ao resultado esperado. Os nomes de sítios e empresas, porventura mencionados, foram utilizados apenas para ilustrar os exemplos, não tendo vínculo nenhum com o livro, não garantindo a sua existência nem divulgação.

Os programas e materiais suplementares estão disponíveis para download em:
www.borgescorporation.blogspot.com

Eventualmente, uma errata poderá ser disponibilizada.

Junho/2012.

Charles dedica:

À minha querida esposa Gisela e a minha pequena Cecília, luzes do meu caminho.

Marco dedica:

À minha querida esposa Tade-Ane e aos meus três especiais filhos, a certeza da continuação da minha existência, Caio, Pedro e Vicente.

“Não queremos apenas deixar um mundo melhor para nossos filhos, queremos muito mais deixar filhos melhores para o mundo”.

AGRADECIMENTOS

É com sinceridade que agradecemos a todas as pessoas que colaboram para o desenvolvimento desta obra:

- Aos colegas do Departamento de Eletrônica do Instituto Federal de Santa Catarina, pelo suporte moral e técnico.

- Aos alunos, sem os quais esta obra não seria possível, tornando visível, com suas necessidades e sugestões, o caminho a seguir.

- Aos amigos, onde a expressão da palavra e dos atos é verdadeira.

- Por último, mas não menos importante: especialmente as nossas esposas e filhos, pelo apoio em todos os momentos e pela compreensão das horas que abdicamos de suas companhias (que não foram poucas), na escrita e desenvolvimento do texto e projetos apresentados.

O DISCÍPULO E O BALAIO

Um discípulo chegou para seu mestre e perguntou:

- Mestre, por que devemos ler e decorar tantas orações se não conseguimos memorizá-las completamente e com o tempo as esquecemos?

O mestre não respondeu imediatamente. Ele ficou olhando para o horizonte e depois ordenou ao discípulo:

- Pegue aquele balaio de junco, desça até o riacho, o encha de água e o traga até aqui.

O discípulo olhou para o balaio, que estava bem sujo, e achou muito estranha a ordem do mestre, mas mesmo assim, obedeceu. Pegou o balaio sujo, desceu os 100 degraus da escadaria até o riacho, encheu o cesto de água e começou a subir de volta.

Como o balaio era todo cheio de furos, a água foi escorrendo e quando o discípulo chegou até o mestre, já não restava mais água nenhuma.

O mestre, então, perguntou:

- Então, meu filho, o que você aprendeu?

O discípulo olhou para o cesto vazio e disse:

- Aprendi que um balaio de junco não segura água.

O mestre ordenou-lhe que repetisse o processo.

Quando o discípulo voltou com o balaio vazio novamente, o mestre perguntou:

- Então, meu filho, e agora, o que você aprendeu?

O discípulo novamente respondeu com sarcasmo:

- Balaio furado não segura água.

O mestre, então, continuou ordenando que o discípulo repetisse a tarefa.

Depois da décima vez, o discípulo estava todo molhado e exausto de tanto descer e subir as escadas. Porém, quando o mestre perguntou de novo:

- Então, meu filho, o que você aprendeu?

O discípulo, olhando para dentro do balaio, percebeu admirado:

- O balaio está limpo! Apesar de não segurar a água, ela acabou por lavá-lo!

O mestre, por fim, concluiu:

- Então meu filho, não importa que você não consiga decorar todas as orações. O que importa, na verdade, é que elas purificam sua mente e sua alma.

Antiga história de um mosteiro chinês.

- O que adianta estudar e estudar, se muito do que estudamos acaba por ser esquecido?

- O estudo melhora o nosso processo cognitivo e nosso raciocínio lógico, preparando-nos mais e mais para enfrentar os problemas da vida. Portanto, comece a passar mais água no seu balaio.

SUMÁRIO

PREFÁCIO	XVII
1. INTRODUÇÃO	1
1.1 ATUALIDADE	1
1.2 MICROPROCESSADORES	3
1.3 TIPOS DE MEMÓRIAS	8
1.4 MICROCONTROLADORES	9
1.5 OS MICROCONTROLADORES AVR	12
1.6 A FAMÍLIA AVR	14
2. O ATMEGA328	17
2.1 AS MEMÓRIAS	22
2.1.1 O REGISTRADOR DE STATUS - SREG	26
2.1.2 O STACK POINTER	28
2.2 DESCRIÇÃO DOS PINOS	30
2.3 SISTEMA DE CLOCK	33
2.4 O RESET	34
2.5 GERENCIAMENTO DE ENERGIA E O MODO SLEEP	37
3. SOFTWARE E HARDWARE	39
3.1 A PLATAFORMA ARDUINO	40
3.2 CRIANDO UM PROJETO NO AVR STUDIO	43
3.3 SIMULANDO NO PROTEUS (ISIS)	49
3.4 GRAVANDO O ARDUINO	52
4. PROGRAMAÇÃO	57
4.1 ALGORITMOS	58
4.1.1 ESTRUTURAS DOS ALGORITMOS	60
4.2 FLUXOGRAMAS	61
4.3 MÁQUINA DE ESTADOS	63
4.4 O ASSEMBLY	64
4.5 PROGRAMAÇÃO C	67
4.5.1 INTRODUÇÃO	69

4.5.2 ESTRUTURAS CONDICIONAIS.....	81
4.5.3 ESTRUTURAS DE REPETIÇÃO	83
4.5.4 DIRETIVAS DE PRÉ-COMPILAÇÃO	87
4.5.5 PONTEIROS	91
4.5.6 VETORES E MATRIZES	91
4.5.7 ESTRUTURAS.....	92
4.5.8 FUNÇÕES	93
4.5.9 O IMPORTANTÍSSIMO TRABALHO COM BITS	95
4.5.10 PRODUZINDO UM CÓDIGO EFICIENTE	101
4.5.11 ESTRUTURAÇÃO E ORGANIZAÇÃO DO PROGRAMA	102
5. PORTAS DE ENTRADA E SAÍDA (I/OS).....	105
5.1 ROTINAS SIMPLES DE ATRASO	107
5.2 LIGANDO UM LED	110
5.3 LENDO UM BOTÃO (CHAVE TÁCTIL).....	114
5.4 ACIONANDO DISPLAYS DE 7 SEGMENTOS	125
5.5 ACIONANDO LCDs 16 x 2	132
5.5.1 INTERFACE DE DADOS DE 8 BITS	133
5.5.2 INTERFACE DE DADOS DE 4 BITS	137
5.5.3 CRIANDO NOVOS CARACTERES	146
5.5.4 NÚMEROS GRANDES	149
5.6 ROTINAS DE DECODIFICAÇÃO PARA USO EM DISPLAYS	154
6. INTERRUPTÕES	157
6.1 INTERRUPTÕES NO ATMEGA328	157
6.2 INTERRUPTÕES EXTERNAS.....	163
6.2.1 UMA INTERRUPTÃO INTERROMPENDO OUTRA.....	170
7. GRAVANDO A EEPROM	173
8. TECLADO MATRICIAL.....	179
9. TEMPORIZADORES/CONTADORES.....	185
9.1 TEMPORIZANDO E CONTANDO	186
9.2 MODULAÇÃO POR LARGURA DE PULSO – PWM	187
9.3 TEMPORIZADOR/CONTADOR 0	189
9.3.1 REGISTRADORES DO TCO	194

9.3.2 CÓDIGOS EXEMPLO	199
9.4 TEMPORIZADOR/CONTADOR 2.....	202
9.4.1 REGISTRADORES DO TC2	203
9.4.2 CÓDIGO EXEMPLO	208
9.5 TEMPORIZADOR/CONTADOR 1.....	211
9.5.1 REGISTRADORES DO TC1	217
9.5.2 CÓDIGOS EXEMPLO	221
9.6 PWM POR SOFTWARE.....	226
9.7 ACIONANDO MOTORES	228
9.7.1 MOTOR DC DE IMÃ PERMANENTE	228
9.7.2 MICRO SERVO MOTOR	231
9.7.3 MOTOR DE PASSO UNIPOLAR.....	234
9.8 MÓDULO DE ULTRASSOM - SONAR.....	238
10. GERANDO MÚSICAS COM O MICROCONTROLADOR	243
11. TÉCNICAS DE MULTIPLEXAÇÃO.....	253
11.1 EXPANSÃO DE I/O MAPEADA EM MEMÓRIA	254
11.2 CONVERSÃO SERIAL-PARALELO	257
11.3 CONVERSÃO PARALELO-SERIAL	260
11.4 MULTIPLEXAÇÃO DE DISPLAYS DE 7 SEGMENTOS	260
11.5 UTILIZANDO MAIS DE UM LED POR PINO	266
11.6 MATRIZ DE LEDS	268
11.7 CUBO DE LEDS.....	271
12. DISPLAY GRÁFICO (128 × 64 PONTOS)	277
12.1 CONVERSÃO DE FIGURAS.....	283
12.2 USANDO O DISPLAY GRÁFICO COMO UM LCD 21 x 8	293
13. FORMAS DE ONDA E SINAIS ANALÓGICOS.....	297
13.1 DISCRETIZANDO UM SINAL	297
13.2 CONVERSOR DIGITAL-ANALÓGICO COM UMA REDE R/2R.....	302
13.3 CONVERSOR DIGITAL-ANALÓGICO COM UM SINAL PWM	305
13.4 SINAL PWM PARA UM CONVERSOR CC-CC (BUCK)	307
13.5 SINAL PWM PARA UM CONVERSOR CC-CA	310
14. SPI.....	317

14.1 SPI DO ATMEGA328	320
14.2 SENSOR DE TEMPERATURA TC72	326
14.3 CARTÃO DE MEMÓRIA SD/MMC	331
15. USART.....	345
15.1 USART DO ATMEGA328	345
15.2 USART NO MODO SPI MESTRE	357
15.3 COMUNICAÇÃO ENTRE O MICROCONTROLADOR E UM COMPUTADOR	358
15.4 MÓDULOS BÁSICOS DE RF	367
15.5 MÓDULO BLUETOOTH PARA PORTA SERIAL	373
16. TWI (TWO WIRE SERIAL INTERFACE) – I2C	377
16.1 I2C NO ATMEGA328.....	379
16.2 REGISTRADORES DO TWI	383
16.3 USANDO O TWI	386
16.4 RELÓGIO DE TEMPO REAL DS1307.....	396
17. COMUNICAÇÃO 1 FIO POR SOFTWARE	411
17.1 SENSOR DE TEMPERATURA DS18S20.....	415
18. COMPARADOR ANALÓGICO	425
18.1 MEDINDO RESISTÊNCIAS E CAPACITÂNCIAS	430
18.2 LENDO MÚLTIPLAS TECLAS	434
19. CONVERSOR ANALÓGICO-DIGITAL - ADC	437
19.1 REGISTRADORES DO ADC.....	444
19.2 MEDIÇÃO DE TEMPERATURA.....	448
19.3 LENDO UM TECLADO MATRICIAL.....	451
19.4 SENSOR DE TEMPERATURA LM35.....	453
19.5 SOBREAMOSTRAGEM.....	456
19.6 FILTRO DE MÉDIA MÓVEL	459
20. MESCLANDO PROGRAMAÇÃO C COM ASSEMBLY	463
20.1 USO DOS REGISTRADORES DO AVR	465
20.2 CONVENÇÃO NA CHAMADA DE FUNÇÕES	465
20.3 EXEMPLOS	466
20.4 O ARQUIVO MAP	470

21. RTOS.....	473
21.1 INTRODUÇÃO	473
21.2 GESTÃO DE TAREFAS	477
21.3 COMUNICAÇÃO E SINCRONIZAÇÃO ENTRE TAREFAS.....	483
21.4 ALOCAÇÃO DINÂMICA DE MEMÓRIA.....	486
21.5 BRTOS	488
22. A IDE DO ARDUINO	499
23. FUSÍVEIS E A GRAVAÇÃO DO ATMEGA328.....	507
23.1 OS FUSÍVEIS E BITS DE BLOQUEIO	508
23.2 HARDWARE DE GRAVAÇÃO	512
23.3 O TEMPORIZADOR WATCHDOG.....	515
23.4 GRAVAÇÃO IN-SYSTEM DO ATMEGA	520
24. CONSIDERAÇÕES FINAIS.....	523
25. REFERÊNCIAS	529

APÊNDICE

A. ASSEMBLY DO ATMEGA.....	537
A.1 INSTRUÇÕES DO ATMEGA	537
A.2 STATUS REGISTER	542
B. DISPLAY DE CRISTAL LÍQUIDO 16 × 2 - CONTROLADOR HD44780	543
B.1 PINAGEM	543
B.2 CÓDIGOS DE INSTRUÇÕES	544
B.3 ENDEREÇO DOS SEGMENTOS (DDRAM)	545
B.4 CONJUNTO E CÓDIGO DOS CARACTERES	545
C. ERROS E AJUSTE DA TAXA DE COMUNICAÇÃO DA USART	547
D. CIRCUITOS PARA O ACIONAMENTO DE CARGAS	551

D.1 A CHAVE TRANSISTORIZADA	551
D.2 CIRCUITOS INTEGRADOS PARA O SUPRIMENTO DE CORRENTES	558
D.3 ACOPLADORES ÓPTICOS	560
E. PROJETO DE PLACAS DE CIRCUITO IMPRESSO - BÁSICO.....	563
E.1 UNIDADE IMPERIAL E MÉTRICA	563
E.2 ENCAPSULAMENTOS	564
E.2.1 COMPONENTES PTH	565
E.2.2 COMPONENTES SMD.....	567
E.3 PADS E VIAS (PTH)	568
E.4 TRILHAS.....	569
E.5 REGRAS BÁSICAS DE DESENHO	571
E.6 PASSOS PARA O DESENHO DA PCI.....	575
E.7 CONCLUSÕES	581
F. TABELAS DE CONVERSÃO	583
G. REGISTRADORES DE I/O DO ATMEGA328	587
H. ACRÔNIMOS.....	609

PREFÁCIO

Aristóteles em sua Metafísica falou que "na verdade, foi pelo espanto que os homens começaram a filosofar tanto no princípio como agora". Todo aquele que inicia o aprendizado dos microcontroladores e se espanta com todas as possibilidades de uso da tecnologia, certamente se tornará um eterno e apaixonado estudante desses dispositivos. A constante evolução dos microcontroladores corrobora os ensinamentos filosóficos de Heráclito de Éfeso, segundo o qual tudo é devir - "não entramos duas vezes em um mesmo rio". A todo instante, somos surpreendidos com o lançamento de novos dispositivos, que superam os seus antecessores em capacidade de processamento e no conjunto de suas aplicações; acompanhar esses avanços requer a disposição para um aprendizado contínuo.

Em 1980, quando a Intel lançou o legendário microcontrolador 8051, que operava a 12 MHz e embarcava 4 kbytes de ROM para receber o código do usuário e 128 bytes de RAM para armazenar dados temporários, não se podia imaginar que uma dezena de microprocessadores ou microcontroladores estariam presentes em nossos lares e carros, na forma de relógios, brinquedos, eletrodomésticos, telefones, computadores de bordo, etc. Fazemos referência ao 8051, porque foi através desse simpático senhor de 32 anos que a nossa geração de técnicos e engenheiros realizaram seus primeiros projetos com microcontroladores. E, ao ingressar na carreira docente, trabalhávamos os conceitos de programação e hardware de sistemas embarcados utilizando a arquitetura MCS-51. Mais tarde, devido à compatibilidade pino a pino entre o 8051 e um dos microcontroladores da família AVR, à época a arquitetura de 8 bits de melhor desempenho, optamos em trabalhar nas disciplinas de microcontroladores com os dispositivos AVR. Com uma pequena modificação na nossa placa desenvolvimento que empregava um microcontrolador AT89S8252, um 8051 da Atmel, começamos a utilizar o

AT90S8515 e, posteriormente, o ATmega8515. Depois, com adaptações e melhorias para o ensino, acabamos desenvolvendo um kit com o ATmega32, incluindo inúmeros periféricos. Atualmente, utilizamos a plataforma Arduino com o ATmega328, auxiliados pelo seu amplo conjunto de módulos de expansão de hardware.

Lançado no ano que o Curso Técnico de Eletrônica do Departamento Acadêmico de Eletrônica do Instituto Federal de Santa Catarina - Campus Florianópolis comemora seu jubileu de prata, este livro, de certa forma, materializa parte das discussões e conhecimentos acumulados no nosso departamento ao longo destes anos e preenche uma lacuna técnica, em língua portuguesa, ao apresentar um grande conjunto de técnicas de projetos com microcontroladores reunidas de forma didática em um único compêndio.

Neste livro, para a descrição do AVR, optou-se pelo Atmega328, que apresenta as principais características desejáveis em um microcontrolador e, ainda, está presente no Arduino Uno, a versão mais popular do módulo de desenvolvimento de código aberto Arduino, plataforma de fácil utilização, inclusive por leigos, e que conta ainda com uma intensa e ampla comunidade global que troca informações e dissemina o seu uso.

A grande maioria dos programas foi testada no Arduino Uno com a utilização de hardware complementar na forma de módulos prontos ou montados em matriz de contato. Como o Arduino é uma plataforma de desenvolvimento fácil de encontrar e barata, possuindo amplo hardware de suporte, fica fácil testar os programas apresentados, permitindo também o rápido desenvolvimento de novos projetos. Os programas foram desenvolvidos no AVR *Studio*® 5.1, o software profissional e gratuito disponibilizado pela Atmel.

Também apresentamos um software para a simulação de microcontroladores (Proteus®), cada vez mais utilizado no meio acadêmico e na indústria, superando a velha prerrogativa do teste do programa em hardware.

Para conhecer os detalhes do microcontrolador, é fundamental a consulta ao manual do fabricante. Aqui apresentamos apenas um resumo e algumas aplicações típicas, ressaltando que as figuras e detalhes técnicos do ATmega foram obtidos desse manual. É importante notar que o fabricante disponibiliza vários *Application Notes* com muitas dicas de projeto, bem como códigos prontos para serem utilizados.

Desenvolvemos os conteúdos para serem ministrados durante um semestre letivo, em cursos com não menos que 4 horas semanais. Entretanto, o conteúdo pode ser dividido em dois semestres de acordo com a carga horária empregada e de acordo com a profundidade no desenvolvimento dos temas apresentados. Isso só é válido para alunos que possuem conhecimento técnico em eletrônica e programação. Todavia, tal conhecimento não é prerrogativa para o estudo deste livro, mas fundamental para a compreensão de muito dos conceitos apresentados.

São propostos inúmeros exercícios ao longo dos capítulos. Eles servem para fixar conceitos e permitir ao aluno aprender novas técnicas, incluindo sugestões de projetos e considerações técnicas. Apesar de ser possível exclusivamente simular os circuitos, o desenvolvimento em hardware não deve ser desprezado, permitindo a conexão entre teoria e prática, aumentando o aprendizado, a capacidade lógica de projeto e a solução de problemas.

Tentamos apresentar os conceitos de forma objetiva para a leitura rápida e fácil. Desta forma, os esquemáticos dos circuitos foram simplificados para facilitar a visualização e compreensão. A complexidade e o conhecimento são gradualmente aumentados ao longo do texto. Após a leitura dos capítulos introdutórios é importante seguir os capítulos ordenadamente. Entretanto, os capítulos de conhecimentos gerais sobre os softwares empregados (capítulo 3) e sobre programação (capítulo 4) seguem uma sequência para melhor organização do conteúdo, podendo ser consultados sempre que necessário.

No capítulo 4 apresentamos uma breve revisão sobre as técnicas necessárias para a programação a fim de suprir eventuais dúvidas e orientar o estudante para a programação microcontrolada. Nesse capítulo, também apresentamos sugestões para o emprego eficiente da linguagem C para o ATmega, conforme recomendado pela Atmel, e explicações para o importantíssimo trabalho com bits.

Quanto à linguagem de programação *assembly*, seu emprego é fundamental para a compreensão da estrutura interna do microcontrolador e sua forma de funcionamento. Pode-se aprender muito com os códigos exemplo apresentados. Todavia, a linguagem C é a principal forma de programação, dada sua portabilidade e facilidade de uso, independente do tipo de microcontrolador empregado. Programar em *assembly* só deve ser considerado para fins de estudo ou quando realmente houver limitações de desempenho e memória do microcontrolador a considerar, por isso empregamos muito pouco o *assembly* e concentramos o foco na programação em linguagem C. Porém, existe um capítulo exclusivo de como utilizar ambos, C e *assembly*, em um único programa, explorando o que há de melhor nas duas linguagens.

No apêndice apresentamos informações relevantes e necessárias para a programação e desenvolvimento de circuitos microcontrolados. Entre elas, estão dois capítulos: um sobre o projeto de chaves transistorizadas, o outro sobre técnicas de desenho para placas de circuito impresso. É importante um contato com esse material quando do projeto completo de circuitos microcontrolados.

Em resumo, este livro é um ponto inicial no desenvolvimento de projetos microcontrolados, cujas técnicas são empregadas para o ensino e projeto. Esperamos, realmente, que ele possa ajudá-lo e inspirá-lo quanto às possibilidades de projetos com microcontroladores.

Bom Estudo!

Os autores.

1. INTRODUÇÃO

Neste capítulo, serão abordados alguns conceitos necessários à compreensão dos microcontroladores, começando com os microprocessadores e as memórias. Posteriormente, será feita uma introdução aos microcontroladores e a apresentação dos mais relevantes disponíveis no mercado, incluindo detalhes sobre os AVR's.

1.1 ATUALIDADE

As melhorias tecnológicas demandam cada vez mais dispositivos eletrônicos. Assim, a cada dia são criados componentes eletrônicos mais versáteis e poderosos. Nessa categoria, os microprocessadores, aos quais os microcontroladores pertencem, têm alcançado grande desenvolvimento. Sua facilidade de uso em amplas faixas de aplicações permite o projeto relativamente rápido e fácil de novos equipamentos. Atualmente, os microcontroladores estão presentes em quase todos os dispositivos eletrônicos controlados digitalmente, como por exemplo: nas casas, em máquinas de lavar, fornos de micro-ondas, televisores, aparelhos de som e imagem, condicionadores de ar e telefones; nos veículos, em sistemas eletrônicos de controle de injeção de combustível, controle de estabilidade, freios ABS (*Anti-lock Braking System*), computadores de bordo e GPS (*Global Positioning System*); nos eletrônicos portáteis, em telefones celulares, tocadores de mídia eletrônica, vídeo games e relógios; na indústria, em controladores lógico programáveis, de motores e fontes de alimentação. Em resumo, os sistemas microcontrolados encontram-se em todos os segmentos eletrônicos, desde simples equipamentos domésticos até sistemas industriais complexos (fig. 1.1).

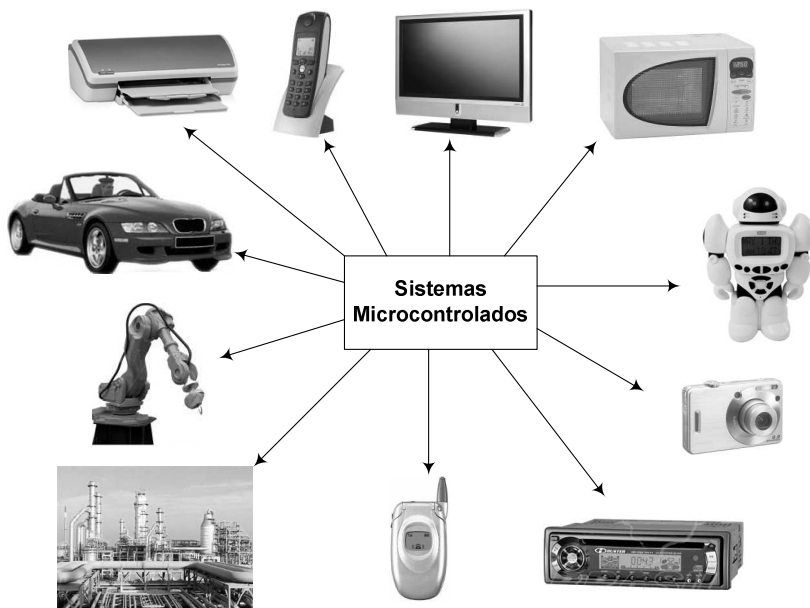


Fig. 1.1 - Sistemas microcontrolados.

O desenvolvimento dos microcontroladores se deve ao grande número de funcionalidades disponíveis em um único circuito integrado. Como o seu funcionamento é ditado por um programa, a flexibilidade de projeto e de formas de trabalho com um hardware específico são inúmeras, permitindo aplicações nas mais diversas áreas.

1.2 MICROPROCESSADORES

Com a evolução da microeletrônica, o nível de miniaturização e o número de funcionalidades dos circuitos integrados evoluíram, permitindo o desenvolvimento de circuitos compactos e programáveis. Assim, surgiram os primeiros microprocessadores.

Um microprocessador é um circuito integrado composto por inúmeras portas lógicas, organizadas de tal forma que permitem a realização de operações digitais, lógicas e aritméticas. Sua operação é baseada na decodificação de instruções armazenadas em uma memória programável. A execução das instruções é ditada por um sinal periódico (relógio - *clock*).

Um diagrama básico de um sistema microprocessado é apresentado na fig. 1.2. O oscilador é responsável pela geração do sinal síncrono de *clock*, o qual faz com que o microprocessador funcione. O coração do sistema é a Unidade Central de Processamento (CPU – *Central Processing Unit*), a qual realiza as operações lógicas e aritméticas exigidas pelo programa. O processador dispõe de duas memórias: uma, onde está o código de programa a ser executado (externa ao chip); e outra, temporária, onde são armazenados informações para o trabalho da CPU (memória de dados). A CPU pode receber e enviar informações ao exterior através da sua interface de entrada e saída. As informações dentro do microprocessador transitam por barramentos, cuja estrutura depende da arquitetura empregada.

Para ler a memória de programa, a CPU possui um contador de endereços (PC – *Program Counter*) que indica qual posição da memória deve ser lida. Após a leitura do código no endereço especificado, o mesmo é decodificado e a operação exigida é realizada. O processo se repete indefinidamente de acordo com o programa escrito.

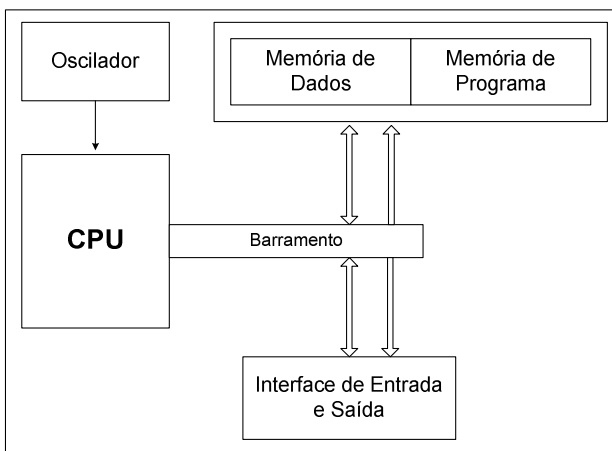


Fig. 1.2 – Estrutura básica de um sistema microprocessado.

Quanto à organização do barramento, existem duas arquiteturas predominantes para as CPUs dos microprocessadores: a arquitetura Von-Neumann, onde existe apenas um barramento interno por onde circulam dados e instruções, e a arquitetura Harvard, caracterizada por dois barramentos internos, um de instruções e outro de dados (Fig. 1.3). Na arquitetura Von-Neumann, a busca de dados e instruções não pode ser executada ao mesmo tempo (gargalo da arquitetura), limitação que pode ser superada com a busca antecipada de instruções (*pipeline*¹) e/ou com caches de instruções/dados. Já na arquitetura Harvard, os dados e instruções podem ser acessados simultaneamente, o que torna essa arquitetura inerentemente mais rápida que a Von-Neumann.

Por sua vez, quanto ao conjunto de instruções, os microprocessadores são classificados em duas arquiteturas: Computadores com Conjunto Complexo de Instruções (CISC – *Complex Instructions Set Computers*) e Computadores com Conjunto Reduzido de Instruções (RISC – *Reduced*

¹ **Pipeline** é uma técnica de hardware que permite a execução em paralelo de operações por parte da CPU. As instruções são colocadas em um fila aguardando o momento de sua execução. Assim, pode-se imaginar um sistema com registradores organizados em cascata e cujo funcionamento é sincronizado por um sinal de *clock*.

Instructions Set Computers). A arquitetura RISC utiliza um conjunto de instruções simples, pequeno e geralmente com extensão fixa. Semelhante a RISC, a arquitetura CISC utiliza um conjunto simples de instruções, porém utiliza também instruções mais longas e complexas, semelhantes às de alto nível, na elaboração de um programa. As instruções CISC são geralmente variáveis em extensão. Assim, a arquitetura RISC necessita de mais linhas de código para executar a mesma tarefa que uma arquitetura CISC, a qual possui muito mais instruções.

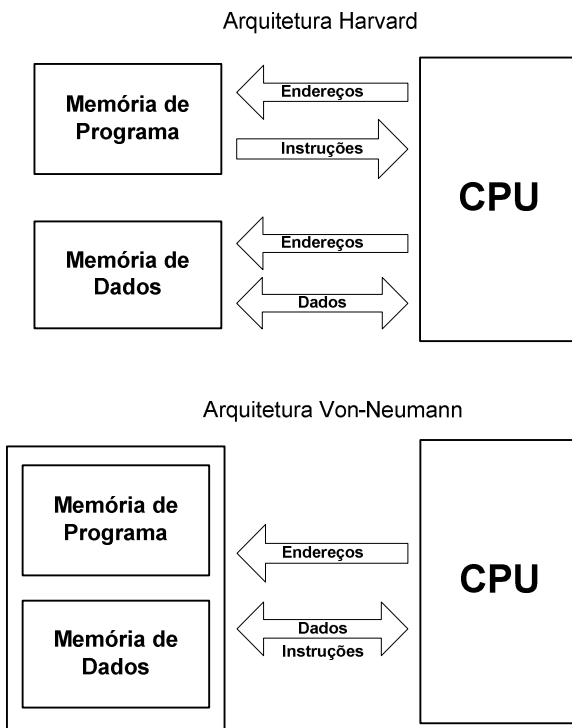


Fig. 1.3 – Arquiteturas clássicas de microprocessadores: Harvard × Von-Neumann.

Muitas literaturas associam a arquitetura Von-Neumann com a arquitetura CISC e a arquitetura Harvard com a arquitetura RISC. A provável razão de tal associação é o fato de muitos microprocessadores

com arquitetura Von-Neumann serem CISC e muitos microprocessadores Harvard serem RISC. Entretanto, a arquitetura CISC e RISC não são sinônimos para as arquiteturas Von-Neumann e Harvard, respectivamente. Por exemplo:

- O núcleo ARM7 é Von-Neumann e RISC.
- O PowerPC é um microprocessador RISC puro, Harvard internamente e Von-Neumann externamente.
- O Pentium-Pro é um microprocessador RISC e Harvard internamente e Von-Neumann e CISC externamente.

A fig. 1.4 é empregada para exemplificar a diferença entre as duas arquiteturas supracitadas para um microprocessador hipotético de 8 bits². O objetivo é multiplicar dois números contidos nos endereços 0 e 3 da memória de um microprocessador hipotético e armazenar o resultado de volta na posição 0.

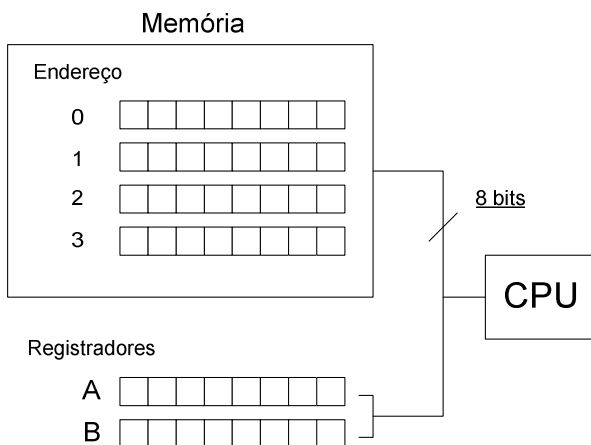


Fig. 1.4 – Diagrama esquemático para a comparação entre um microprocessador CISC e um RISC.

² Para um detalhamento completo ver: CARTER, Nicholas, *Computer Architecture*. 1 ed. New York: McGraw Hill, 2002.

Em microprocessadores CISC, o objetivo é executar a tarefa com o menor número de códigos possíveis (*assembly*). Assim, um microprocessador CISC hipotético poderia ter a seguinte instrução:

```
MULT 0,3      //multiplica o conteúdo do endereço 0 com o do endereço 3
               //armazena o resultado no endereço 0.
```

Para um microprocessador RISC, a resolução do problema seria feita por algo como:

```
LOAD  A,0      //carrega o registrador A com o conteúdo do endereço 0
LOAD  B,3      //carrega o registrador B com o conteúdo do endereço 3
MULT  A,B      //multiplica o conteúdo de A com o de B, o resultado fica em A
STORE 0,A      //armazena o valor de A no endereço 0
```

O microprocessador RISC emprega o conceito de carga e armazenamento (*Load and Store*) utilizando registradores de uso geral. Os dados a serem multiplicados necessitam primeiro ser carregados nos registradores apropriados (A e B) e, posteriormente, o resultado é armazenado na posição de memória desejada (0). Em um microprocessador CISC, o dado pode ser armazenado sem a necessidade do uso explícito de registradores.

Esse conceito também pode ser visto quando se deseja escrever diretamente nos pinos de saída do microprocessador. Considerando que se deseja alterar os níveis lógicos (0 e 1) entre 8 pinos do microprocessador, expressos pela variável P1, o código resultante seria:

CISC:

```
MOV P1,0xAA    //escreve diretamente nos 8 pinos (P1) o valor binário 10101010
```

RISC:

```
LOAD A,0xAA    //carrega o registrador A com o valor binário 10101010
OUT  P1,A      //escreve o valor de A nos 8 pinos (P1)
```

Em um microprocessador RISC, antes de serem movidos para outro local, os valores sempre precisam passar pelos registradores de uso geral, por isso a caracterização da arquitetura como *Load and Store*.

Apesar dos microprocessadores RISC empregarem mais códigos para executar a mesma tarefa que os CISCs, eles são mais rápidos, pois geralmente executam uma instrução por ciclo de *clock*, ao contrário dos CISCs, que levam vários ciclos de *clock*. Todavia, o código em um microprocessador RISC será maior.

Os processadores RISC empregam mais transistores nos registradores de memória, enquanto que os CISC fazem uso de um grande número deles no seu conjunto de instruções.

O emprego de registradores nas operações de um processador RISC é uma vantagem, pois os dados podem ser utilizados posteriormente sem a necessidade de recarga, reduzindo o trabalho da CPU.

1.3 TIPOS DE MEMÓRIAS

Em sistemas digitais existem inúmeros tipos de memórias e, constantemente, novas são criadas para suprir a crescente demanda por armazenagem de informação. Basicamente, um sistema microprocessado contém duas memórias: a memória de programa, que armazenará o código a ser executado e a memória de dados, onde os dados de trabalho da CPU podem ser escritos e lidos.

Antigamente, a memória de programa era uma memória que só podia ser lida (ROM - *Read Only Memory*, memória somente de leitura) e que era gravada uma única vez (as memórias de gravação única se chamam OTP - *One Time Programmable*, programável somente uma vez). As memórias que eram regraváveis utilizavam raios ultravioleta para o seu apagamento e o chip possuía uma janela de quartzo para tal apagamento. Hoje em dia, se empregam memórias de programa apagáveis eletricamente, permitindo inúmeras gravações e regravagens de forma rápida e sem a necessidade de equipamentos especiais.

Atualmente, nos microcontroladores são empregadas dois tipos de memórias regraváveis eletricamente: a memória *flash* EEPROM para o programa, e a *standart* EEPROM (*Electrical Erasable Programming Read Only Memory*) para armazenamento de dados que não devem ser perdidos. A diferença básica entre elas é que a memória *flash* só pode ser apagada por setores (vários bytes de uma única vez) e, na EEPROM, os bytes são apagados individualmente de forma mais lenta.

Para a armazenagem de dados que não precisam ficar retidos após a desenergização do circuito, se emprega uma memória volátil, chamada memória RAM (*Random Access Memory*, memória de acesso aleatório). Em microcontroladores, a memória usual é a SRAM (*Static RAM*), sendo fundamental para a programação, pois a maioria dos programas necessita de variáveis temporárias. Uma distinção importante é que a memória RAM não é a memória de dados da CPU, e sim uma memória para armazenagem temporária de dados do programa do usuário.

1.4 MICROCONTROLADORES

Um microcontrolador é o um sistema microprocessado com várias funcionalidades (periféricos) disponíveis em um único chip. Basicamente, um microcontrolador é um microprocessador com memórias de programa, de dados e RAM, temporizadores e circuitos de *clock* embutidos. O único componente externo que pode ser necessário é um cristal para determinar a frequência de trabalho. A grande vantagem de se colocar várias funcionalidades em um único circuito integrado é a possibilidade de desenvolvimento rápido de sistemas eletrônicos com o emprego de um pequeno número de componentes.

Dentre as funcionalidades encontradas nos microcontroladores, pode-se citar: gerador interno independente de *clock* (não necessita de cristal ou componentes externos); memória SRAM, EEPROM e *flash*; conversores analógicos-digitais (ADCs), conversores digitais-analógicos (DACs); vários

temporizadores/contadores; comparadores analógicos; saídas PWM; diferentes tipos de interface de comunicação, incluindo USB, USART, I2C, CAN, SPI, JTAG, Ethernet; relógio de tempo real; circuitos para gerenciamento de energia no chip; circuitos para o controle de inicialização (*reset*); alguns tipos de sensores; interface para LCD; e outros periféricos de acordo com o fabricante. Na fig. 1.5, é apresentado um diagrama esquemático de um microcontrolador típico.

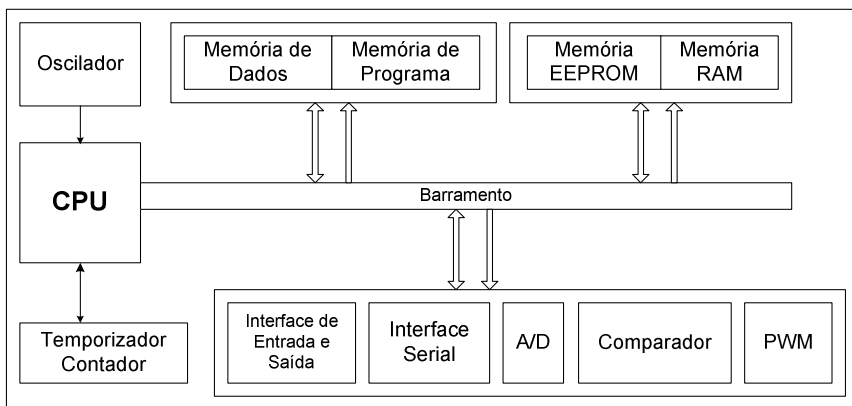


Fig. 1.5 – Diagrama esquemático de um microcontrolador típico.

Na tab. 1.1, é apresentada uma lista das várias famílias dos principais fabricantes de microcontroladores da atualidade. A coluna Núcleo indica o tipo de arquitetura ou unidade de processamento que constitui a base do microcontrolador. Dependendo do fabricante, existem ambientes de desenvolvimento que podem ser baixados gratuitamente da internet.

Tab. 1.1 – Principais fabricantes e famílias de microcontroladores (08/2011).

Fabricante	Sítio na Internet	Barram.	Família	Arquit.	Núcleo
Analog Device	www.analog.com	8 bits	ADuC8xx	CISC	8051
		32 bits	ADuC7xx	RISC	ARM7
		32 bits	ADuCRF101	RISC	Cortex M3
Atmel	www.atmel.com	8 bits	AT89xx	CISC	8051
		8 bits	AVR (ATtiny – ATmega – Xmega)	RISC	AVR8
		32 bits	AT32xx	RISC	AVR32
		32 bits	SAM7xx	RISC	ARM7
		32 bits	SAM9xx	RISC	ARM9
		32 bits	SAM3xx	RISC	Cortex M3
Cirrus Logic	www.cirrus.com	32 bits	EP73xx	RISC	ARM7
		32 bits	EP93xx	RISC	ARM9
Silicon Labs	www.silabs.com	8 bits	C8051xx	CISC	8051
Freescale	www.freescale.com	8 bits	RS08xx	CISC	
		8 bits	HCS08xx	CISC	6808
		8 bits	HC08xx - HC05 - HC11	CISC	
		16 bits	S12x - HC12 - HC16	CISC	
		32 bits	ColdFire	CISC	
		32 bits	68K	CISC	68000
		32 bits	K10-K70	RISC	Cortex M4
		32 bits	MPC5xxx	RISC	Power Arch
		32 bits	PXxxx	RISC	Power Arch
Fujitsu	www.fujitsu.com	8 bits	MB95xx	CISC	F ² MC-8FX
		16 bits	MB90xx	CISC	F ² MC-16FX
		32 bits	MB91xx	RISC	FR
		32 bits	FM3	RISC	Cortex M3
Holtek	www.holtek.com	8 bits	HT4x/5x/6x/8x/9x	RISC	
		8 bits	HT85F22x	CISC	8051
		32 bits	HT32F125x	RISC	Cortex M3
Infineon	www.infineon.com	8 bits	C500 - XC800	CISC	8051
		16 bits	C116 - XC166 - XC2x00	CISC	C116
		32 bits	TC11x	RISC	TriCore Arch
Maxim	www.maxim-ic.com	8 bits	DS 80/83/87/89xx	CISC	8051
		16 bits	MAXQxx	RISC	MAXQ20
		32 bits	ZA9Lx	RISC	ARM9
		32 bits	MAXQxx	RISC	MAXQ30
Microchip	www.microchip.com	32 bits	USIP	RISC	MIPS 4kSD
		8 bits	PIC 10/12/16/18	RISC	
		16 bits	PIC24 - dsPIC	RISC	
National Semic.	www.national.com	32 bits	PIC32	RISC	MIPS32 M4K
		8 bits	COP8	CISC	COP8
NXP	www.nxp.com	8 bits	80C51	CISC	8051
		16 bits	XA	CISC	
		32 bits	LPC 21/22/23/2400	RISC	ARM7
		32 bits	LPC 29/31/3200	RISC	ARM9
		32 bits	LPC 11/1200	RISC	Cortex M0
		32 bits	LPC 13/17/1800	RISC	Cortex M3
		32 bits	LPC 4300	RISC	Cortex M4

Fabricante	Sítio na Internet	Barram.	Família	Arquit.	Núcleo
Renesas	www.renesas.com	16 bits	RL78 / 78k /R8C/M16C	CISC	
		32 bits	M32C - H8S- RX	CISC	
		32 bits	V850 - SuperH	RISC	SH
ST	www.st.com	8 bits	STM8	RISC	STM8A
		8 bits	ST6/7	CISC	
		16 bits	ST10	CISC	
		32 bits	STR7	RISC	ARM7
		32 bits	STR9	RISC	ARM9
		32 bits	STM32F/W	RISC	Cortex M3
Texas Instruments	www.ti.com	16 bits	MSP430	RISC	
		32 bits	C2000	RISC	
		32 bits	Stellaris ARM	RISC	Cortex M3
		32 bits	TMS570	RISC	Cortex R4F
Toshiba	www.toshiba.com/taec	8 bits	TMP 86/87/88/89xx	CISC	
		16/32 bits	TMP 91/92/93/94/95xx	CISC	
		32 bits	TMPA9xx	RISC	ARM9
		32 bits	TMPM3xx	RISC	Cortex M3
Zilog	www.zilog.com	8 bits	Z8xx/Z8Encore/eZ80xx	CISC	Z80
		16 bits	ZNEO (Z16xx)	CISC	

Atualmente, nas arquiteturas modernas de microcontroladores há o domínio da Harvard/RISC, a qual evoluiu para a chamada arquitetura RISC avançada ou estendida. Ao contrário da RISC tradicional, essa é composta por um grande número de instruções, que utilizam uma quantidade reduzida de portas lógicas, produzindo um núcleo de processamento compacto, veloz e que permite uma programação eficiente (gera um menor número de linhas de código). Devido às questões de desempenho, compatibilidade eletromagnética e economia de energia, é importante que um microcontrolador execute a maioria das instruções em poucos ciclos de *clock*, diminuindo o consumo e a dissipação de energia.

1.5 OS MICROCONTROLADORES AVR

A história dos microcontroladores AVR começa em 1992 na Norwegian *University of Science and Technology* (NTNU), na Noruega, onde dois estudantes de doutorado, Alf-Egil-Boden e Vegard Wollan, defenderam uma tese sobre um microcontrolador de 8 bits com memória de programa *flash* e arquitetura RISC avançada. Dos seus nomes surgiu a sigla AVR: Alf – Vegard – RISC. Percebendo as vantagens do microcontrolador que haviam

criado, Alf e Vegard refinaram o seu projeto por dois anos. Com o AVR aperfeiçoado, foram para a Califórnia (EUA) para encontrar um patrocinador para a ideia, a qual foi comprada pela Atmel, que lançou comercialmente o primeiro AVR em meados de 1997, o AT90S1200. Desde então, o aperfeiçoamento do AVR é crescente, juntamente com a sua popularidade.

O AVR apresenta uma boa eficiência de processamento e um núcleo compacto (poucos milhares de portas lógicas). O desempenho do seu núcleo de 8 bits é superior ao da maioria das outras tecnologias de 8 bits disponíveis atualmente no mercado³. Com uma arquitetura RISC avançada, o AVR apresenta mais de uma centena de instruções e uma estrutura voltada à programação C, permitindo a produção de códigos compactos (menor número de bytes por funcionalidade). Também, apresenta inúmeros periféricos que o tornam adequado para uma infinidade de aplicações.

As principais características dos microcontroladores AVR são:

- executam a maioria das instruções em 1 ou 2 ciclos de *clock* (poucas em 3 ou 4) e operam com tensões entre 1,8 V e 5,5 V, com velocidades de até 20 MHz. Estão disponíveis em diversos encapsulamentos;
- alta integração e grande número de periféricos com efetiva compatibilidade entre toda a família AVR;
- possuem vários modos para redução do consumo de energia e características adicionais (*picoPower*) para sistemas críticos;
- possuem 32 registradores de propósito geral e instruções de 16 bits (cada instrução ocupa 2 bytes na memória de programa);
- memória de programação *flash* programável *in-system*, SRAM e EEPROM;
- facilmente programados e com *debug in-system* via interface simples, ou com interfaces JTAG compatível com 6 ou 10 pinos;
- um conjunto completo e gratuito de softwares;
- preço acessível.

³ Recentemente a ST Microelectronics lançou o STM8, um forte concorrente, em termos de desempenho, do AVR de 8 bits.

Existem microcontroladores AVR específicos para diversas áreas, tais como: automotiva, controle de LCDs, redes de trabalho CAN, USB, controle de motores, controle de lâmpadas, monitoração de bateria, 802.15.4/ZigBee™ (comunicação sem fio, *wireless*) e controle por acesso remoto.

1.6 A FAMÍLIA AVR

Dentre os principais componentes da família AVR, pode-se citar:

- **tinyAVR® - ATtiny** – 4 até 28 pinos de I/O.
Microcontroladores de propósito geral de até 8 kbytes de memória *flash*, 512 bytes de SRAM e EEPROM.
- **megaAVR® - ATmega** – 23 até 86 pinos de I/O.
Microcontroladores com vários periféricos, multiplicador por hardware, até 256 kbytes de memória *flash*, 4 kbytes de EEPROM e 8 kbytes de SRAM.
- **picoPower™ AVR**. Microcontroladores com características especiais para economia de energia (são designados com a letra P, como por exemplo, ATmega328P).
- **XMEGA™ ATxmega** – 50 até 78 pinos de I/O.
Os microcontroladores XMEGA dispõem de avançados periféricos para o aumento de desempenho, tais como: DMA (*Direct Memory Access*) e sistema de eventos.
- **AVR32** (não pertence às famílias acima) – 28 até 160 pinos de I/O.
Microcontroladores de 32 bits com arquitetura RISC projetada para maior processamento por ciclos de *clock*, com eficiência de 1,3 mW/MHz e até 210 DMIPS (Dhrystone *Million Instructions per Second*⁴) a 150 MHz, conjunto de instruções para DSP (*Digital Signal Processing*, processamento digital de sinais) com SIMD (*Single Instruction Multiple Data*, instrução simples dados múltiplos), soluções SoC (*System-on-a-chip*, sistema em um chip) e suporte completo ao Linux.

As principais características para alguns dos AVRs ATtiny e ATmega são apresentadas nas tabs. 1.2 e 1.3. Para aplicações que exijam outras funcionalidades de hardware além das apresentadas, o sítio do fabricante deve ser consultado.

⁴ **Dhrystone** é um tipo de programa, chamado *benchmark*, desenvolvido para testar o desempenho de processadores. O DMIPS é obtido quando a pontuação obtida do Dhrystone é dividida por 1,757 (o número de Dhrystone por segundo obtida de uma máquina nominal de 1 MIPS – 1 Milhão de Instruções por Segundo).

Tab. 1.2 – Comparação entre alguns ATtiny*.

<u>ATtiny</u>	<i>Flash</i> (Kbytes)	EEPROM (Bytes)	SRAM (Bytes)	Max Pinos I/O	Canais AD 10-bits	16-bit TC	8-bit TC	Canais PWM	SPI	TWI	F.max (MHz)	VCC (V)
ATtiny10	1	--	32	4	--	1	--	2	--	--	12	1,8 - 5,5
ATtiny12	1	64	--	6	--	--	1	--	--	--	8	1,8 - 5,5
ATtiny13A	1	64	64	6	4	--	1	2	--	--	20	1,8 - 5,5
ATtiny2313	2	128	128	18	--	1	1	4	USI	USI	20	1,8 - 5,5
ATtiny24	2	128	128	12	8	1	1	4	USI	USI	20	1,8 - 5,5
ATtiny24A	2	128	128	12	8	1	1	4	USI	USI	20	1,8 - 5,5
ATtiny25	2	128	128	6	4	--	2	4	USI	USI	20	1,8 - 5,5
ATtiny261A	2	128	128	16	11	1	2	2	sim	USI	20	1,8 - 5,5
ATtiny28L	2	--	32	11	--	--	1	--	--	--	4	1,8 - 5,5
ATtiny4	0,5	--	32	4	--	1	--	2	--	--	12	1,8 - 5,5
ATtiny43U	4	64	256	16	4	--	2	4	sim	--	8	0,7 - 1,8
ATtiny44A	4	256	256	12	8	1	1	4	USI	USI	20	1,8 - 5,5
ATtiny45	4	256	256	6	4	--	2	4	USI	USI	20	1,8 - 5,5
ATtiny461A	4	256	256	16	11	1	2	2	sim	USI	20	1,8 - 5,5
ATtiny48	4	64	256	24/28	6/8	1	1	1	sim	sim	12	1,8 - 5,5
ATtiny5	0,5	--	32	4	--	1	--	2	--	--	12	1,8 - 5,5
ATtiny84	8	512	512	12	8	1	1	4	USI	USI	20	1,8 - 5,5
ATtiny85	8	512	512	6	4	--	2	4	USI	USI	20	1,8 - 5,5
ATtiny861A	8	512	512	16	11	1	2	2	sim	USI	20	1,8 - 5,5
ATtiny88	8	64	512	24/28	6/8	1	1	1	sim	sim	12	1,8 - 5,5
ATtiny9	1	--	32	4	--	1	--	2	--	--	12	1,8 - 5,5

*Os ATtiny possuem oscilador interno RC e comparador analógico.

Tab. 1.3 – Comparação entre alguns ATmega*.

ATmega	Flash (Kbytes)	EEPROM (Kbytes)	SRAM (Bytes)	Max Pinos I/O	Canais AD 10-bits	16bit TC	8-bit TC	Canais PWM	SPI	I ² C	USART	F _{max} (MHz)	VCC (V)
Atmega1280	128	4	8192	86	16	4	2	16	1+USART	sim	4	16	1,8 – 5,5
Atmega1281	128	4	8192	54	8	4	2	9	1+USART	sim	2	16	1,8 – 5,5
Atmega1284P	128	4	16K	32	8	2	2	6	sim	sim	2	20	1,8 – 5,5
Atmega128A	128	4	4096	53	8	2	2	8	1	sim	2	16	2,7 – 5,5
Atmega162	16	0,5	1024	35	--	2	2	6	1	--	2	16	1,8 – 5,5
Atmega164P	16	0,5	1024	32	8	1	2	6	1+USART	sim	2	20	1,8 – 5,5
Atmega165P	16	0,5	1024	54	8	1	2	4	1+USI	USI	1	16	1,8 – 5,5
Atmega168PA	16	0,5	1024	23	8	1	2	6	1+USART	sim	1	20	1,8 – 5,5
Atmega16A	16	0,5	1024	32	8	1	2	4	1	sim	1	16	2,7 – 5,5
Atmega2560	256	4	8192	86	16	4	2	16	1+USART	sim	4	16	1,8 – 5,5
Atmega2561	256	4	8192	54	8	4	2	9	1+USART	sim	2	16	1,8 – 5,5
Atmega324P	32	1	2048	32	8	1	2	6	1+USART	sim	2	20	1,8 – 5,5
Atmega3250	32	1	2048	69	8	1	2	4	1+USI	USI	1	16	1,8 – 5,5
Atmega3250P	32	1	2048	69	8	1	2	4	1+USI	USI	1	20	1,8 – 5,5
Atmega325P	32	1	2048	54	8	1	2	4	1+USI	USI	1	20	1,8 – 5,5
Atmega328P	32	1	2048	23	8	1	2	6	1+USART	sim	1	20	1,8 – 5,5
Atmega32A	32	1	2048	32	8	1	2	4	1	sim	1	16	2,7 – 5,5
Atmega48PA	4	0,25	512	23	8	1	2	6	1+USART	sim	1	20	1,8 – 5,5
Atmega64	64	2	4096	54	8	2	2	8	1	sim	2	16	2,7 – 5,5
Atmega640	64	4	8192	86	16	4	2	16	1+USART	sim	4	16	1,8 – 5,5
Atmega644	64	2	4096	32	8	1	2	6	1+USART	sim	1	20	1,8 – 5,5
Atmega645	64	2	4096	54	8	1	2	4	1+USI	USI	1	16	1,8 – 5,5
Atmega6450	64	2	4096	69	8	1	2	4	1+USI	USI	1	16	1,8 – 5,5
Atmega64A	64	2	4096	54	8	2	2	8	1	sim	2	16	2,7 – 5,5
Atmega8515	8	0,5	512	35	--	1	1	3	1	--	1	16	2,7 – 5,5
Atmega8535	8	0,5	512	32	8	1	2	4	1	sim	1	16	2,7 – 5,5
Atmega88PA	8	0,5	1024	23	8	1	2	6	1+USART	sim	1	20	1,8 – 5,5
Atmega8A	8	0,5	1024	23	6/8	1	2	3	1	sim	1	16	2,7 – 5,5

* Os ATmega possuem oscilador interno RC, comparador analógico, RTc (*Real Time counter*) e multiplicador por hardware.