MC404

ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

2010

Prof. Paulo Cesar Centoducatte
Prof. Mario Lúcio Côrtes
Prof. Ricardo Pannain

MC404

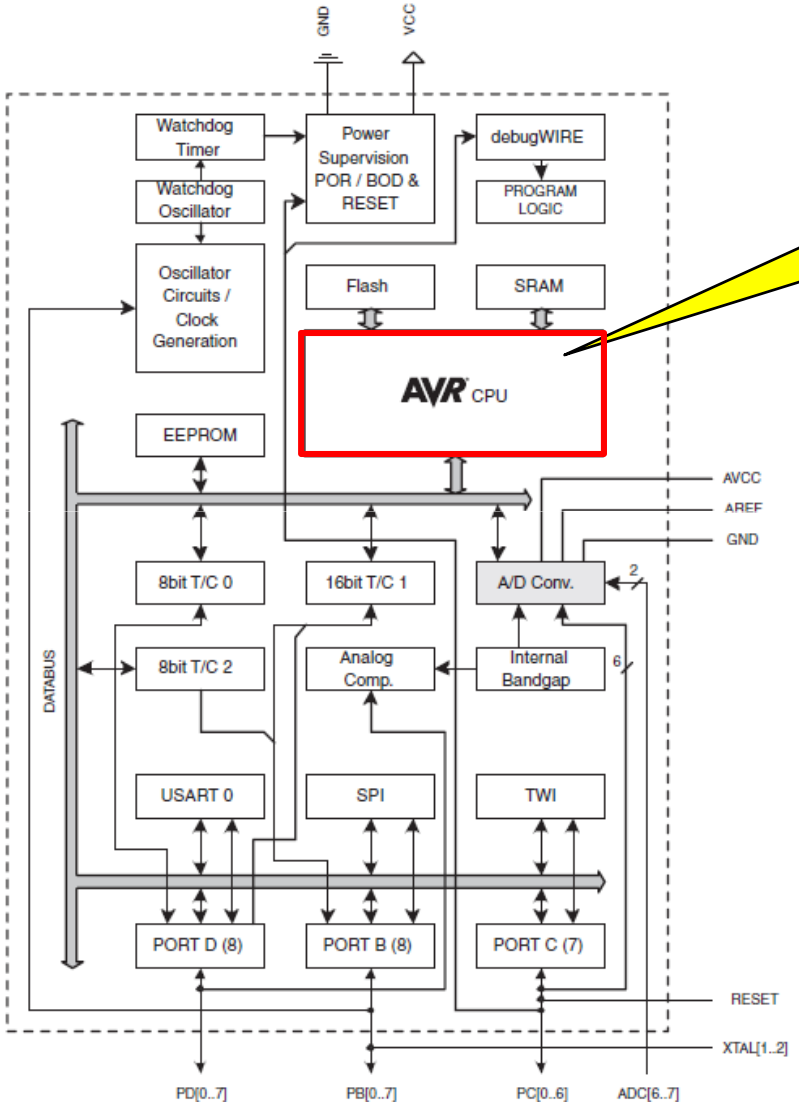
ORGANIZAÇÃO BÁSICA DE COMPUTADORES E LINGUAGEM DE MONTAGEM

"Interrupção e modos de sleep"

Interrupção
Sumário

- Conceitos de interrupção
- Os vetores de interrupção
- Tipos de interrupção
- Os controles de interrupção: habilitação, máscaras e flags
- Temporizadores
- Modos de sleep

Comunicação e E/S



CPU deve interagir com eventos on-chip (timers, usart etc) e off-chip (eventos de entrada e saída)

Possibilidades:

- **Programa em loop (busy wait) monitorando continuamente sinal externo**
- **Polling: programa consulta cíclicamente várias possíveis requisições, de forma intercalada com trabalho "útil"**
- **Interrupção: programa executa normalmente suas tarefas e é interrompido por evento externo**

Imp ... o AVR

26 vetores

Evento externo→CPU executa instrução nesta posição de memória

2 Interrupt Vectors in ATmega8...

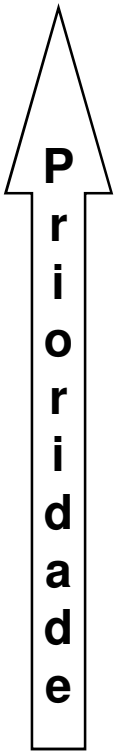
...ble 9-2. Reset and Interrupt ...tors in ATmega88

Vector No.	Program Address[2]	Source	Interrupt Definition
1	0x000[1]	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x001	INT0	External Interrupt Request 0
3	0x002	INT1	External Interrupt Request 1
4	0x003	PCINT0	Pin Change Interrupt Request 0
5	0x004	PCINT1	Pin Change Interrupt Request 1
6	0x005	PCINT2	Pin Change Interrupt Request 2
7	0x006	WDT	Watchdog Time-out Interrupt
8	0x007	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x008	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x009	TIMER2 OVF	Timer/Counter2 Overflow
11	0x00A	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x00B	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x00C	TIMER1 COMPB	Timer/Coutner1 Compare Match B
14	0x00D	TIMER1 OVF	Timer/Counter1 Overflow
15	0x00E	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x00F	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x010	TIMER0 OVF	Timer/Counter0 Overflow
18	0x011	SPI, STC	SPI Serial Transfer Complete
19	0x012	USART, RX	USART Rx Complete
20	0x013	USART, UDRE	USART, Data Register Empty
21	0x014	USART, TX	USART, Tx Complete
22	0x015	ADC	ADC Conversion Complete
23	0x016	EE READY	EEPROM Ready
24	0x017	ANALOG COMP	Analog Comparator
25	0x018	TWI	2-wire Serial Interface
26	0x019	SPM READY	Store Program Memory Ready

Possíveis fontes de interrupção

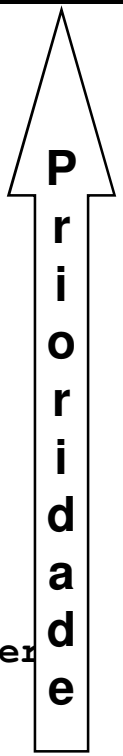
Programa Típico (1)

```
Address Labels   Code              Comments
0x000    rjmp RESET                ; Reset Handler
0x001    rjmp EXT_INT0             ; IRQ0 Handler
0x002    rjmp EXT_INT1             ; IRQ1 Handler
0x003    rjmp PCINT0               ; PCINT0 Handler
0x004    rjmp PCINT1               ; PCINT1 Handler
0x005    rjmp PCINT2               ; PCINT2 Handler
0x006    rjmp WDT                  ; Watchdog Timer Handler
0x007    rjmp TIM2_COMPA           ; Timer2 Compare A Handler
0X008    rjmp TIM2_COMPB           ; Timer2 Compare B Handler
0x009    rjmp TIM2_OVF             ; Timer2 Overflow Handler
0x00A    rjmp TIM1_CAPT            ; Timer1 Capture Handler
0x00B    rjmp TIM1_COMPA           ; Timer1 Compare A Handler
0x00C    rjmp TIM1_COMPB           ; Timer1 Compare B Handler
0x00D    rjmp TIM1_OVF             ; Timer1 Overflow Handler
0x00E    rjmp TIM0_COMPA           ; Timer0 Compare A Handler
0x00F    rjmp TIM0_COMPB           ; Timer0 Compare B Handler
```



Programa Típico (2)

```
Address Labels   Code     Comments
.................
0x010    rjmp TIM0_OVF             ; Timer0 Overflow Handler
0x011    rjmp SPI_STC              ; SPI Transfer Complete Handler
0x012    rjmp USART_RXC            ; USART, RX Complete Handler
0x013    rjmp USART_UDRE           ; USART, UDR Empty Handler
0x014    rjmp USART_TXC            ; USART, TX Complete Handler
0x015    rjmp ADC                  ; ADC Conversion Complete Handler
0x016    rjmp EE_RDY               ; EEPROM Ready Handler
0x017    rjmp ANA_COMP             ; Analog Comparator Handler
0x018    rjmp TWI                  ; 2-wire Serial Interface Handler
0x019    rjmp SPM_RDY              ; Store Program Memory Ready Handler
;
0x01A RESET:
        ldi r16, high(RAMEND)      ; Main program start
0x01B   out SPH,r16                ; Set Stack Pointer to top of RAM
0x01C   ldi r16, low(RAMEND)
0x01D   out SPL,r16
0x01E   sei                        ; Enable interrupts
0x01F   <instr> xxx
```



Algumas interrupções

- **Vetor 1 (posição $0) → interrupção de RESET**
- **Vetores 2 e 3 (posições $1 e $2) →interrupções externas INT0 e INT1 geradas nos pinos 4 e 5 da CPU, respectivamente (ver pag 2 ou)**
- **Vetores 15 a 17 → "Timer 0"**
- **Vetor 23 (posição $16) → fim de escrita da EEPROM**
- **Prioridade em solicitação simultânea:**
  - **menores endereços → maiores prioridades: 0x000 (RESET) tem a maior prioridade e 0x019 (SPM_RDY) tem a menor**

Habilitação e ativação da interrupção

- **Habilitação global (afeta a todos os tipos de interrupção): bit 7 (bit I) do SREG.**
  - **Set = habilita (instrução SEI) e Clear = desabilita (instrução CLI)**
- **Habilitação por tipo de interrupção: bit específico em registradores de E/S chamados de "máscara de interrupções" (EIMSK, TIMSK2, TIMSK1, TIMSK0, PCMSK2, PCMSK1, PCMSK0)**
- **Exemplo – para habilitar interrupções INT0 ou INT1, geradas por sinal nos pinos 4 ou 5 da CPU:**
  - **Ativar bits 0 ou 1 do registrador EIMSK (External Interrupt Mask Register) e/**
  - **Habilitar globalmente → bit 7 em 1 do SREG**
- **Ativação: para cada tipo de interrupção → bit para registrar o "pedido da interrupção" gerado pelo evento causador**
  - **Chamados de "flags", localizados em registradores de E/S com nomes do tipo "... Flag Register"**
  - **Por exemplo, bits 0 ou 1 do EIFR (External Interrupts Flag Register) registram os pedidos de interrupção INT0 ou INT1**

Mecanismo de interrupção (sequência)

- **Endereço da próxima instrução → pilha**
- **Interrupções são desabilitadas globalmente, clear bit 7 do SREG, e localmente, clear bit de pedido de interrupção (flag no registrador apropriado)**
- **A prórpia rotina de interrupção deve salvar contexto, SREG e eventuais registradores que serão alterados**
- **Concluído o tratamento da interrupção, retorno ao programa interrompido (instrução RETI)**
  - **desempilha e coloca no PC o endereço da próxima instrução a ser executada**
  - **liga o bit 7 no SREG (habilitando globalmente interrupções)**
  - **Nenhuma interrupção é aceita antes que a próxima instrução seja executada**

Definição do tipo de sinal gerador de interrupção

- **Interrupções externas: INT0 ou INT1 (EXT_INT0 e EXT_INT1)**
- **Tipo de sinal definido no EICRA – External Interrupt Control Register A**
- **Bits 3-2 (INT1) e bits 1-0 (INT0)**

Bit	7	6	5	4	3	2	1	0
	–	–	–	–	ISC11	ISC10	ISC01	ISC00
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

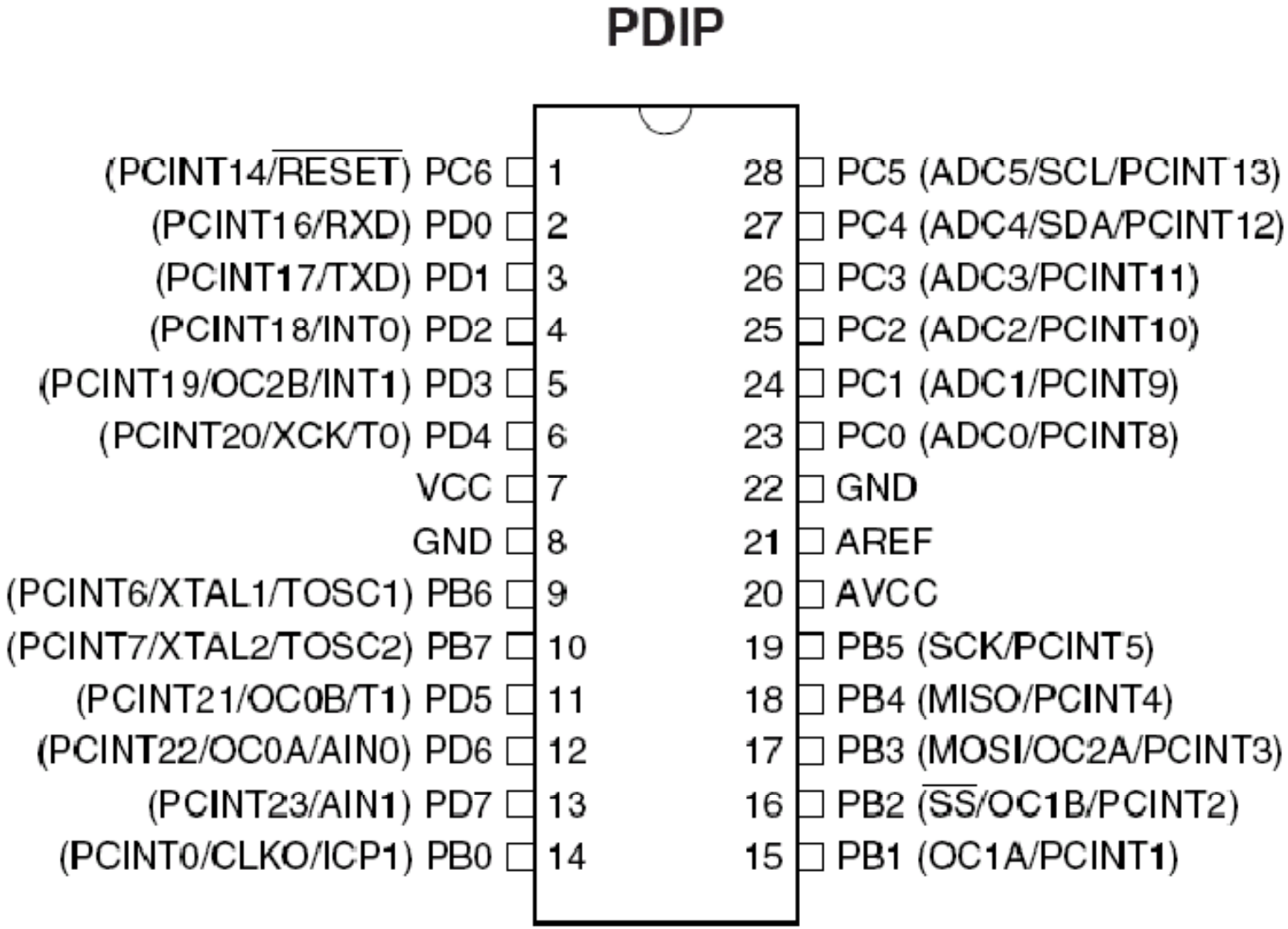
Table 11-1. Interrupt 1 Sense Control

ISC11	ISC10	Description
0	0	The low level of INT1 generates an interrupt request.
0	1	Any logical change on INT1 generates an interrupt request.
1	0	The falling edge of INT1 generates an interrupt request.
1	1	The rising edge of INT1 generates an interrupt request.

Outras fontes de interrupção externa

- **Os 23 pinos de E/S do Atmega88 também pode ser configurado para gerar uma interrupção quando há uma mudança de sinal ("toggle") no pino**
- **Pinos PCINT0..23, agrupados em 3 conjuntos**
- **Vetores de interrupção: PCINT0, PCINT1, PCINT2**
- **Máscara: PCMSK0, PCMSK1 e PCMSK2 (p. 336)**
- **Habilitação da interrupção:bits 0 a 2 do registrador de controle PCICR são usados para habilitar as interrupções correspondentes PCIE0, PCIE1 e PCIE2;**
- **Flag: bits 0 a 2 do registrador de flag PCIFR → pedidos de interrupção de cada um dos 3 grupos**
  - **possível identificar o grupo que interrompeu (pelo flag)**
  - **se dois sinais ativos dentro do mesmo grupo → impossível identificar**

Pinout do ATmega 88



Fontes de interrupção

Reset →

Interrupção externa →

Interrupção externa adicional →

Temporizadores →

Outros →

Source	Interrupt Definition
RESET	External Pin, Power-on Reset, Brown-out R
INT0	External Interrupt Request 0
INT1	External Interrupt Request 1
PCINT0	Pin Change Interrupt Request 0
PCINT1	Pin Change Interrupt Request 1
PCINT2	Pin Change Interrupt Request 2
WDT	Watchdog Time-out Interrupt
TIMER2 COMPA	Timer/Counter2 Compare Match A
TIMER2 COMPB	Timer/Counter2 Compare Match B
TIMER2 OVF	Timer/Counter2 Overflow
TIMER1 CAPT	Timer/Counter1 Capture Event
TIMER1 COMPA	Timer/Counter1 Compare Match A
TIMER1 COMPB	Timer/Coutrer1 Compare Match B
TIMER1 OVF	Timer/Counter1 Overflow
TIMER0 COMPA	Timer/Counter0 Compare Match A
TIMER0 COMPB	Timer/Counter0 Compare Match B
TIMER0 OVF	Timer/Counter0 Overflow
SPI, STC	SPI Serial Transfer Complete
USART, RX	USART Rx Complete
USART, UDRE	USART, Data Register Empty
USART, TX	USART, Tx Complete
ADC	ADC Conversion Complete
EE READY	EEPROM Ready
ANALOG COMP	Analog Comparator
TWI	2-wire Serial Interface
SPM READY	Store Program Memory Ready

Temporizadores (exemplo Timer/Counter0)

- **Contador de 8 bits (registrador TCNT0)**
- **Frequência de contagem:**
  - **Clock dividido pelo "prescaler" (1, 8, 64, 256 e 1024**
  - **Definição do prescaler: bits 0, 1 e 2 do registrador TCCR0B Timer/Counter 0 Control Register B**

Table 12-9. Clock Select Bit Description

CS02	CS01	CS00	Description
0	0	0	No clock source (Timer/Counter stopped)
0	0	1	$clk_{I/O}$/(No prescaling)
0	1	0	$clk_{I/O}$/8 (From prescaler)
0	1	1	$clk_{I/O}$/64 (From prescaler)
1	0	0	$clk_{I/O}$/256 (From prescaler)
1	0	1	$clk_{I/O}$/1024 (From prescaler)
1	1	0	External clock source on T0 pin. Clock on falling edge.
1	1	1	External clock source on T0 pin. Clock on rising edge.

Interrupções geradas pelo Timer/Counter0

- **Pode gerar interrupção quando:**
  - **contagem = valor A**
  - **contagem = valor B**
  - **contador → overflow**
- **Máscaras em TIMSK0**
  - **Bit 2 – OCIE0B: Timer/Counter Output Compare Match B Interrupt Enable**
  - **Bit 1 – OCIE0A: Timer/Counter0 Output Compare Match A Interrupt Enable**
  - **Bit 0 – TOIE0: Timer/Counter0 Overflow Interrupt Enable**

Timer/Counter Interrupt Mask Register – TIMSK0

7	6	5	4	3	2	1	0
–	–	–	–	–	OCIE0B	OCIE0A	TOIE0

Contagem = valor

Output Compare Register A – OCR0A

Bit	7	6	5	4	3	2	1	0	
	OCR0A[7:0]								OCR0A
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

The Output Compare Register A contains an 8-bit value that is continuously compared with the counter value (TCNT0). A match can be used to generate an Output Compare interrupt, or to generate a waveform output on the OC0A pin.

Output Compare Register B – OCR0B

Bit	7	6	5	4	3	2	1	0
	OCR0B[7:0]							
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Registrador de flag do Timer/Counter0

- **Bit 2 – OCF0B: Timer/Counter 0 Output Compare B Match Flag**
- **Bit 1 – OCF0A: Timer/Counter 0 Output Compare A Match Flag**
- **Bit 0 – TOV0: Timer/Counter0 Overflow Flag**

Timer/Counter 0 Interrupt Flag Register – TIFR0

Bit	7	6	5	4	3	2	1	0	
	–	–	–	–	–	OCF0B	OCF0A	TOV0	TIFR0
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Inital Value	0	0	0	0	0	0	0	0	

Esqueleto de rotina de tratamento de interrupção

```
push r16          ; vamos usar r16 para salvar SREG na pilha
in r16, SREG
push r16          ; SREG salvo: nenhuma das instruções acima alterou SREG
push rxx          ; salvo rxx para usar na rotina
. . .             ; salva outros registradores, se preciso
. . .             ; código da rotina de interrupção
pop rxx           ; preparando para sair, pops devem ser dados na ordem inversa
pop r16           ; r16 tem agora o valor original do SREG
out SREG, r16     ; restauramos o SREG original
pop r16           ; e o r16
                  ; agora o topo da pilha tem o endereço de retorno,
reti              ; volta ao programa interrompido, ligando o bit I em SREG
```

Modos de sleep

- **AVR tem 5 modos de sleep**
- **Sleep: desativa seletivamente módulos do AVR que utilizam o clock → consomem mais energia**
- **Habilitação: bit SE (bit 0) do registrador SMCR**
- **Seleção de modo: bits 1-3 do registrador SMCR**

4	3	2	1	0
–	SM2	SM1	SM0	SE

Sleep Mode Control Register – SMCR

Table 7-1. Sleep Mode Select

SM2	SM1	SM0	Sleep Mode
0	0	0	Idle
0	0	1	ADC Noise Reduction
0	1	0	Power-down
0	1	1	Power-save
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Standby(1)
1	1	1	Reserved

Alguns modos: após instrução sleep

- **Idle (SM2, SM1 e SM0 do SMCR = 000):**
  - **CPU para**
  - **relógios continuam funcionando: temporizadores, Watchdog, USART, ADC**
  - **interrupções externas continuam habilitadas**
  - **. Pode-se, no entanto, desativar vários relógios antes de executar sleep através do registrador PRR conforme visto acima.**
- **Power-down (SM2, SM1 e SM0 do SMCR = 010):**
  - **oscilador externo para**
  - **Para acordar a CPU: interrupções externas (INT0 ou INT1), interrupção do Watchdog Timer, da interface serial de 2 pinos ou um reset físico**
  - **Consumo= 0.6 uA a 25ºC**
- **Consumo de potência: tensão de alimentação, frequência, módulos ativos**

Desligar módulos selecionados: PRR

Power Reduction Register - PRR

7	6	5	4	3	2	1	0
PRTWI	PRTIM2	PRTIM0	–	PRTIM1	PRSPI	PRUSART0	PRADC

- **Possível desligar módulos selecionados**
  - **Bit 7 - PRTWI: Power Reduction TWI**
  - **Bit 6 - PRTIM2: Power Reduction Timer/Counter2**
  - **Bit 5 - PRTIM0: Power Reduction Timer/Counter0**
  - **Bit 4 - Res: Reserved bit**
  - **Bit 3 - PRTIM1: Power Reduction Timer/Counter1**
  - **Bit 2 - PRSPI: Power Reduction Serial Peripheral Interface**
  - **Bit 1 - PRUSART0: Power Reduction USART0**
  - **Bit 0 - PRADC: Power Reduction ADCG**

Exemplo: entrando em sleep

- **Para que a instrução sleep tenha efeito: ativar bit SE (sleep enable) do registrador SMCR**
- **Exemplo de código:**

```
ldi r16, $5
out SMCR, r16     ; liga SE, habilita modo power down
sei               ; habilita globalmente interrupções
sleep             ; para a CPU, aguarda interrupção permitida pelo modo
....              ; executa a partir daqui após retornar da rotina de interrupção
```