



Universidade Federal de Santa Maria

Departamento de Eletrônica e Computação

## PROJETO DE SISTEMAS EMBARCADOS

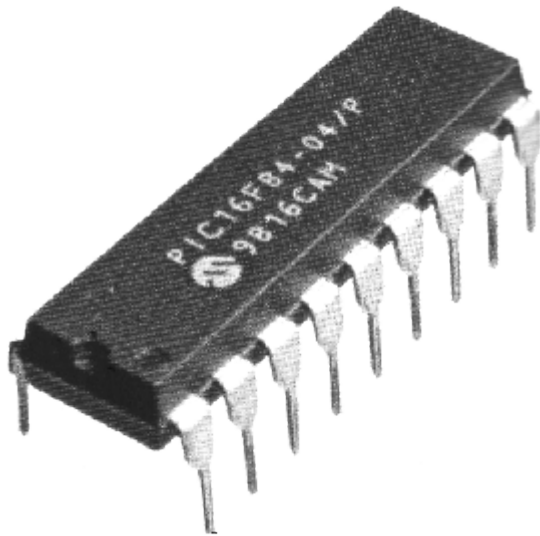
Acesso ao hardware em  
microcontroladores

Prof. Carlos Henrique Barriquello  
barriquello@gmail.com

# Objetivos

---

- Compreender como é feita a interface entre microcontrolador e o mundo externo



# Tema e conteúdo

---

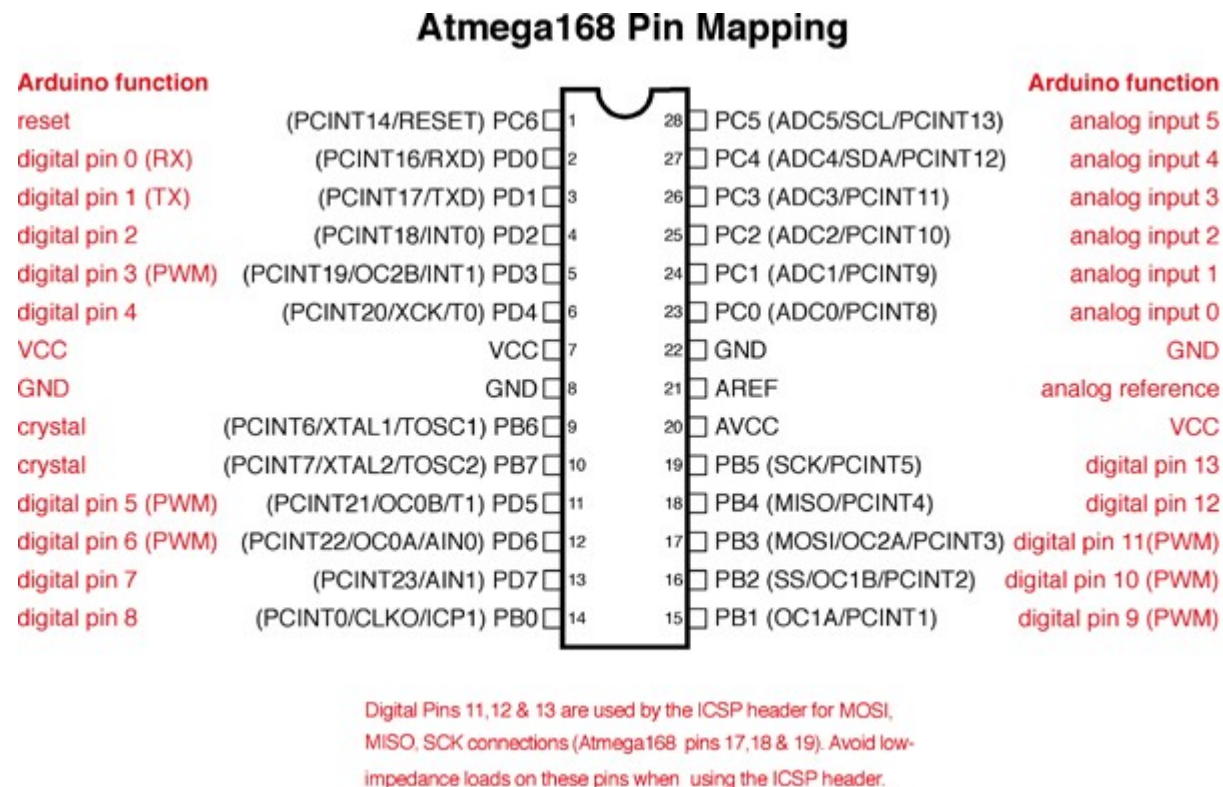
- Modos de E/S (I/O).
- Dispositivos periféricos.
- Interrupções.

# Portas de E/S digitais

---

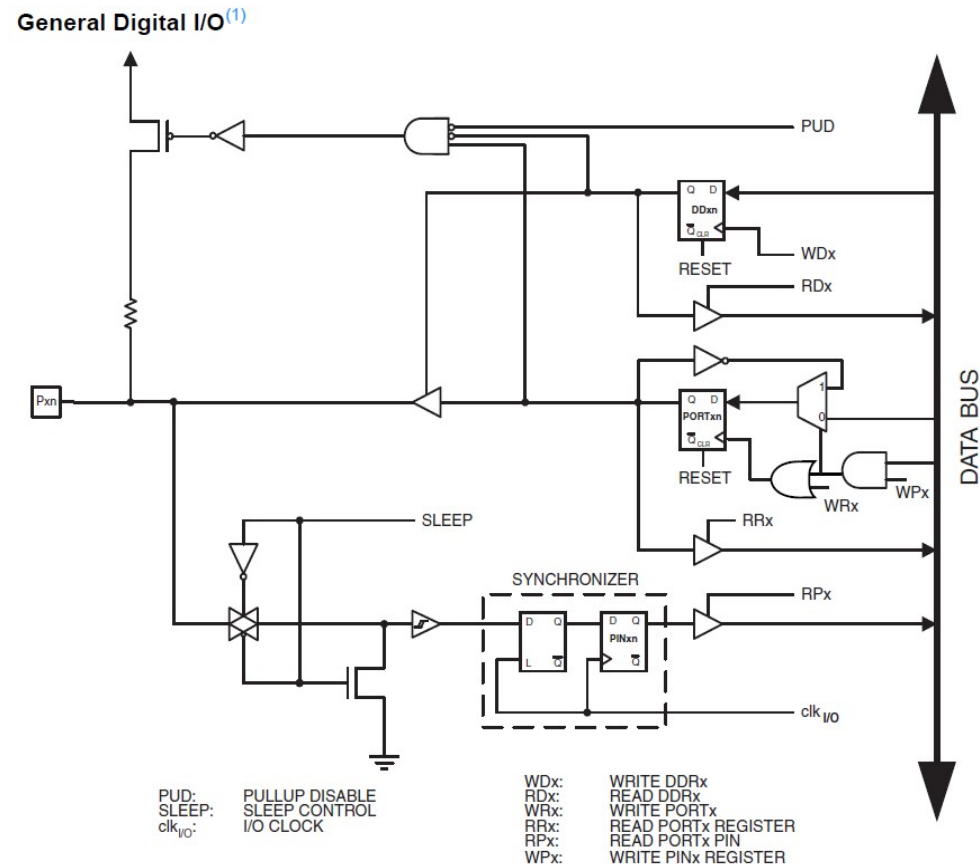
- Pinos como saída digital:
  - Ligar/desligar
  - Acionamento de relés
  - Acionamento de LEDs
  - Display de 7 segmentos
- Pinos como entrada digital:
  - Leitura de teclas/botões (teclado)
  - Sensor de dois estados (ligado/desligado)

# Portas de E/S digitais



!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# Portas de E/S digitais



!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# Portas de E/S digitais

**Table 14-1.** Port Pin Configurations

DDxn	PORTxn	PUD (in MCUCR)	I/O	Pull-up	Comment
0	0	X	Input	No	Tri-state (Hi-Z)
0	1	0	Input	Yes	Pxn will source current if ext. pulled low.
0	1	1	Input	No	Tri-state (Hi-Z)
1	0	X	Output	No	Output Low (Sink)
1	1	X	Output	No	Output High (Source)

## MCUCR – MCU Control Register

Bit	7	6	5	4	3	2	1	0	
0x35 (0x55)	–	BODS <sup>(1)</sup>	BODSE <sup>(1)</sup>	PUD	–	–	IVSEL	IVCE	MCUCR
Read/Write	R	R/W	R/W	R/W	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**PUD: Pull-up Disable**

!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# Portas de E/S digitais

## PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
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	

## DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0	DDRB
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	

## PINB – The Port B Input Pins Address<sup>(1)</sup>

Bit	7	6	5	4	3	2	1	0	
0x03 (0x23)	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0	PINB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A	

!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>



# Portas de E/S digitais

---

- Exemplos:

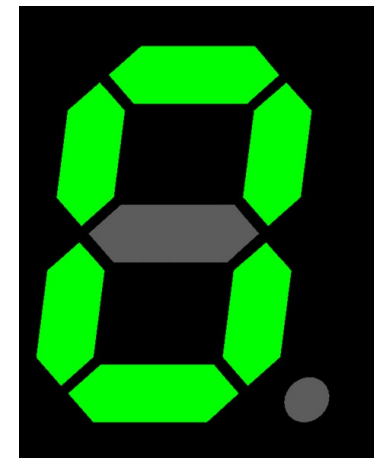
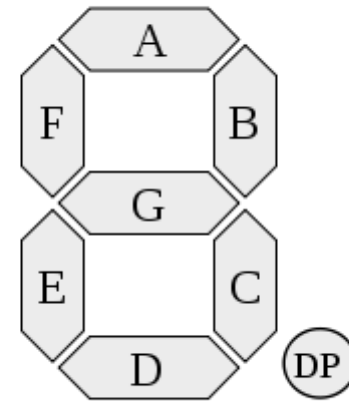
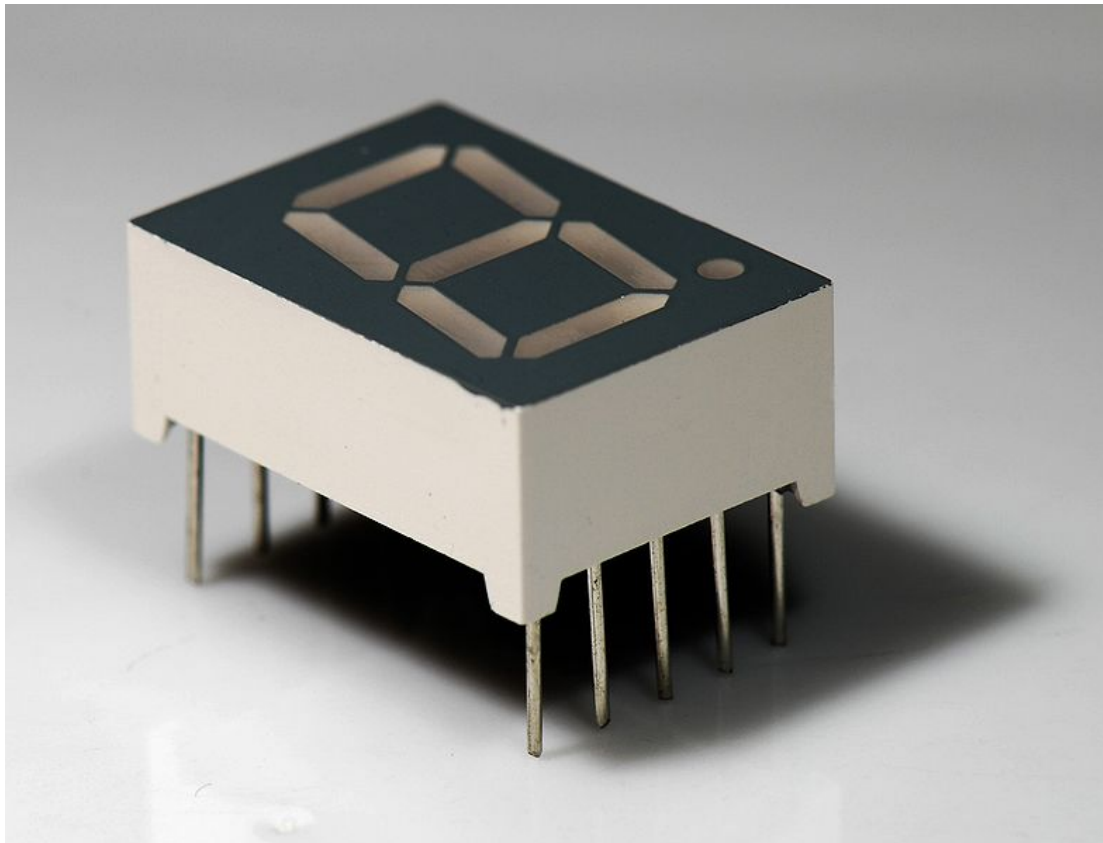
```
DDRD = 0xff;    /* porta como saída
PORTD = 0xff;
PORTD = 0;
PORTD |= (1 << PORTD0);
PORTD &= ~(1 << PORTD1);
PIND = (1 << PIND0) | (1 << PIND1);
```

!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# Displays (mostradores)

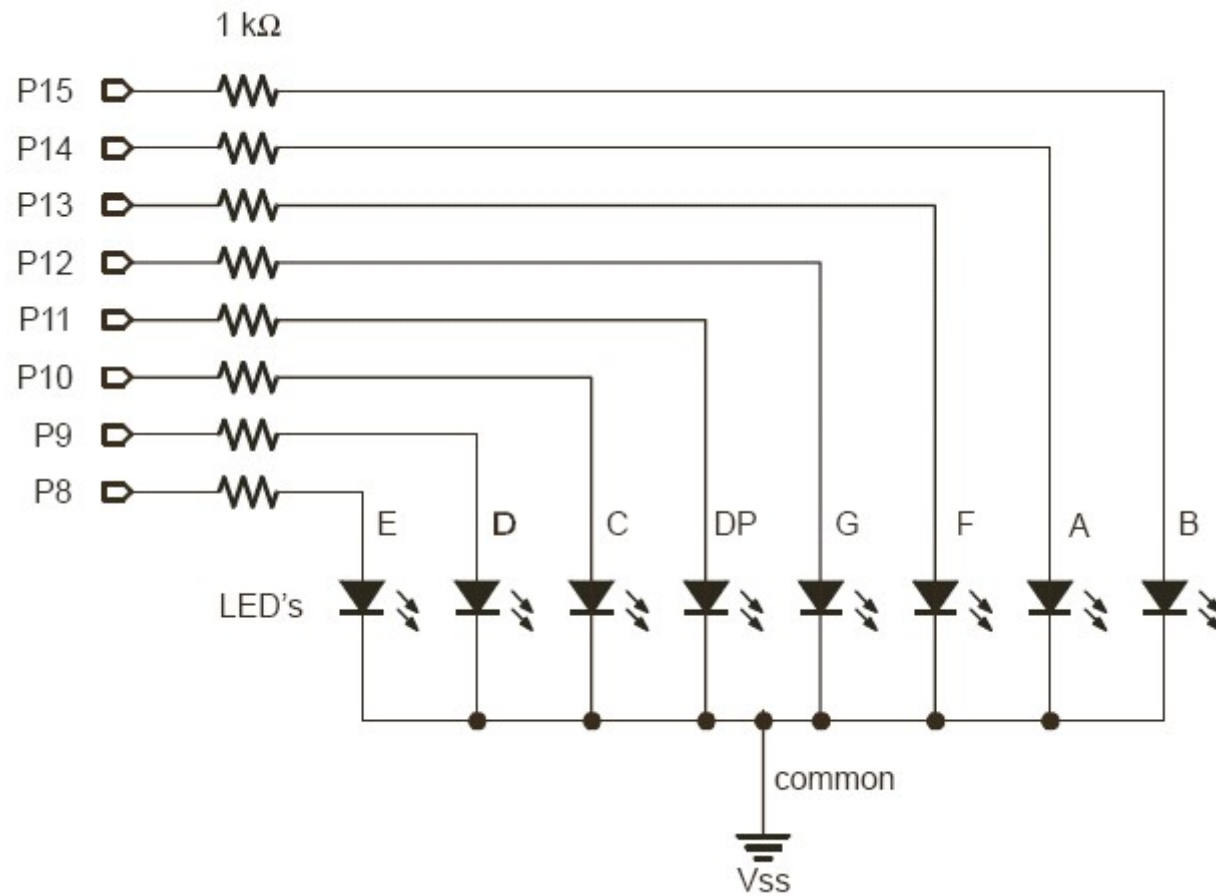
---

- Display de 7 segmentos
  - Uso em mostradores numéricos e alfanuméricos



# Displays (mostradores)

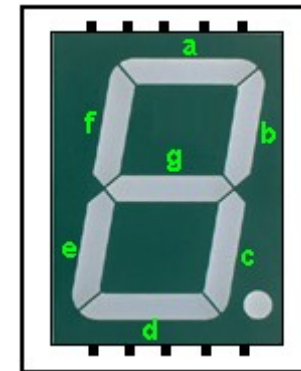
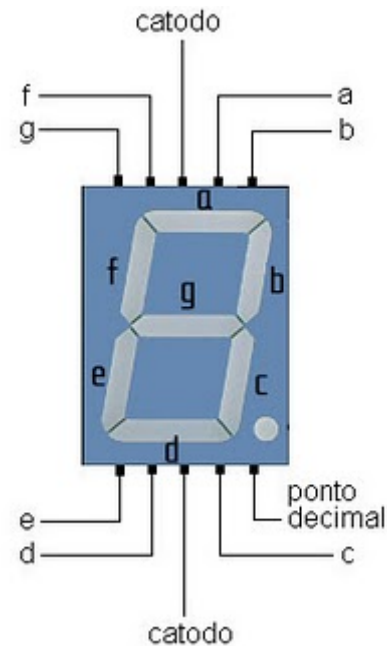
---



# Displays (mostradores)

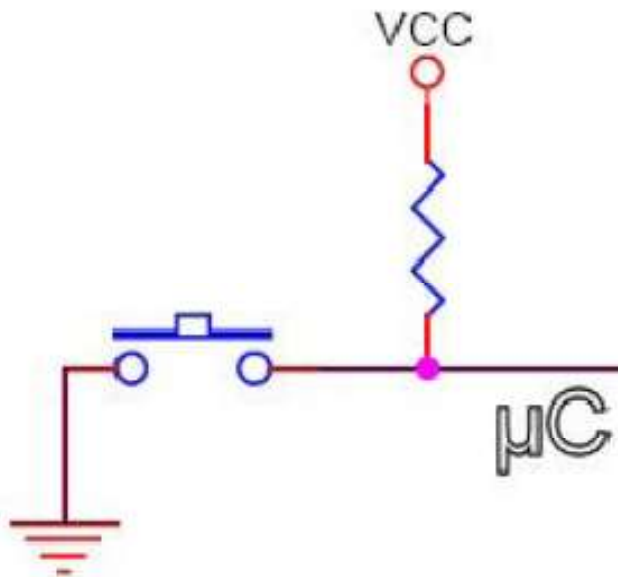
- Cada segmento é controlado por um pino

entradas BCD				segmentos de saída							DISPLAY
D	C	B	A	a	b	c	d	e	f	g	
0	0	0	0	1	1	1	1	1	1	0	0
0	0	0	1	0	1	1	0	0	0	0	1
0	0	1	0	1	1	0	1	1	0	1	2
0	0	1	1	1	1	1	1	0	0	1	3
0	1	0	0	0	1	1	0	0	1	1	4
0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	1	1	1	1	0	0	0	0	7
1	0	0	0	1	1	1	1	1	1	1	8
1	0	0	1	1	1	1	0	0	1	1	9
1	1	1	1	0	0	0	0	0	0	0	



# Tecla/botão

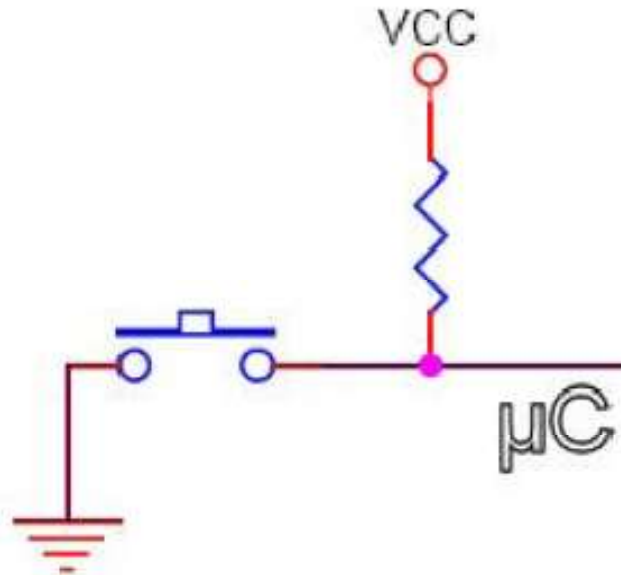
---



- As chaves mecânicas normalmente apresentam a configuração demonstrada ao serem conectadas ao pino de um microcontrolador

# Tecla/botão

---



- O resistor de *pull-up* força um nível lógico alto no pino do microcontrolador configurado como entrada.
- Caso este resistor não seja utilizado, o pino do microcontrolador fica sujeito a ruídos.

# Dois modos de funcionamento

---

– *Polling*

*X*

– Interrupção

# Polling

---

- *Polling:*
  - O processador permanece verificando os registradores de estado do periférico para detectar alterações de estado.
  - Pode ser feito sempre, esporadicamente ou periodicamente



# Interrupção

---

- Interrupção
  - O periférico informa ao processador alterações de estado de seus registradores através do controlador de interrupções.
  - O processador é “interrompido” e desvia o fluxo de execução do programa, se as interrupções estão **habilitadas**.

# Interrupção

---

- Eventos síncronos ou assíncronos podem ocorrer durante a execução das instruções.
- Esses eventos são, em geral, mais prioritários do que as instruções sendo executadas.
- Por isso, o microcontrolador deve interromper o que está fazendo e executar o código de tratamento do evento.
- Esses eventos são chamados de **interrupções** ou **exceções**.

# Interrupções

---

- O código de tratamento da interrupção (ISR) possui um endereço na memória.
- Quando ocorrer o evento (interrupção), o microcontrolador:
  - interrompe a execução do código atual
  - salva seu estado na pilha (memória RAM)
  - e passa a executar o código de tratamento

## VETOR DE INTERRUPÇÃO:

o endereço do código de tratamento deve ser armazenado em um local reservado, conhecido como **vetor de interrupção**.

# Vetores de interrupção

Os **vetores de interrupção** são, em geral, armazenados em uma região no final dos endereços de memória.

0xFFD6:0xFFD7	ACMP	Vacmp
0xFFD8:0xFFD9	ADC Conversion	Vadc
0xFFDA:0xFFDB	KBI Interrupt	Vkeyboard
0xFFDC:0xFFDD	Reserved	—
0xFFDE:0xFFDF	SCI Transmit	Vscitx
0xFFE0:0xFFE1	SCI Receive	Vscirx
0xFFE2:0xFFE3	SCI Error	Vscierr
0xFFE4:0xFFE5	Reserved	—
0xFFE6:0xFFE7	MTIM Overflow	Vmtim
0xFFE8:0xFFE9	Reserved	—
0xFFEA:0xFFEB	Reserved	—
0xFFEC:0xFFED	Reserved	—
0xFFEE:0xFFEF	Reserved	—
0xFFFF0:0xFFFF1	TPM Overflow	Vtpmovf
0xFFFF2:0xFFFF3	Reserved	—
0xFFFF4:0xFFFF5	Reserved	—
0xFFFF6:0xFFFF7	TPM Channel 0	Vtpmcho
0xFFFF8:0xFFFF9	Low Voltage Detect or Low Voltage Warning	Vlvd
0xFFFFA:0xFFFFB	IRQ	Virq
0xFFFFC:0xFFFFD	SWI	Vswi
0xFFFFE:0xFFFFF	Reset	Vreset

!!! Exemplo referente a um microcontrolador HCS08 da NXP –

[www.nxp.com/assets/documents/data/en/reference-manuals/HCS08RMV1.pdf](http://www.nxp.com/assets/documents/data/en/reference-manuals/HCS08RMV1.pdf)

# Vetores de interrupção

No mínimo, **um vetor de interrupção** deve ser sempre programado. Este é o vetor de **RESET**.

0xFFD6:0xFFD7	ACMP	Vacmp
0xFFD8:0xFFD9	ADC Conversion	Vadc
0xFFDA:0xFFDB	KBI Interrupt	Vkeyboard
0xFFDC:0xFFDD	Reserved	—
0xFFDE:0xFFDF	SCI Transmit	Vscitx
0xFFE0:0xFFE1	SCI Receive	Vscirx
0xFFE2:0xFFE3	SCI Error	Vscierr
0xFFE4:0xFFE5	Reserved	—
0xFFE6:0xFFE7	MTIM Overflow	Vmtim
0xFFE8:0xFFE9	Reserved	—
0xFFEA:0xFFEB	Reserved	—
0xFFEC:0xFFED	Reserved	—
0xFFEE:0xFFEF	Reserved	—
0xFFFF0:0xFFFF1	TPM Overflow	Vtpmovf
0xFFFF2:0xFFFF3	Reserved	—
0xFFFF4:0xFFFF5	Reserved	—
0xFFFF6:0xFFFF7	TPM Channel 0	Vtpmcho
0xFFFF8:0xFFFF9	Low Voltage Detect or Low Voltage Warning	Vlvd
0xFFFFA:0xFFFFB	IRQ	Virq
0xFFFFC:0xFFFFD	SWI	Vswi
0xFFFFE:0xFFFFF	Reset	Vreset

!!! Exemplo referente a um microcontrolador HCS08 da NXP –

[www.nxp.com/assets/documents/data/en/reference-manuals/HCS08RMV1.pdf](http://www.nxp.com/assets/documents/data/en/reference-manuals/HCS08RMV1.pdf)

!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# Vetores de interrupção

VectorNo.	Program Address <sup>(2)</sup>	Source	Interrupt Definition
1	0x0000 <sup>(1)</sup>	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter2 Overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x0016	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 Compare Match B
14	0x001A	TIMER1 OVF	Timer/Counter1 Overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x0020	TIMER0 OVF	Timer/Counter0 Overflow
18	0x0022	SPI, STC	SPI Serial Transfer Complete
19	0x0024	USART, RX	USART Rx Complete
20	0x0026	USART, UDRE	USART, Data Register Empty
21	0x0028	USART, TX	USART, Tx Complete
22	0x002A	ADC	ADC Conversion Complete
23	0x002C	EE READY	EEPROM Ready
24	0x002E	ANALOG COMP	Analog Comparator
25	0x0030	TWI	2-wire Serial Interface
26	0x0032	SPM READY	Store Program Memory Ready

Os **vetores de interrupção** são, em geral, armazenados em uma região no início ou no final dos endereços de memória.

!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# Vetores de interrupção

VectorNo.	Program Address <sup>(2)</sup>	Source	Interrupt Definition
1	0x0000 <sup>(1)</sup>	RESET	External Pin, Power-on Reset, Brown-out Reset and Watchdog System Reset
2	0x0002	INT0	External Interrupt Request 0
3	0x0004	INT1	External Interrupt Request 1
4	0x0006	PCINT0	Pin Change Interrupt Request 0
5	0x0008	PCINT1	Pin Change Interrupt Request 1
6	0x000A	PCINT2	Pin Change Interrupt Request 2
7	0x000C	WDT	Watchdog Time-out Interrupt
8	0x000E	TIMER2 COMPA	Timer/Counter2 Compare Match A
9	0x0010	TIMER2 COMPB	Timer/Counter2 Compare Match B
10	0x0012	TIMER2 OVF	Timer/Counter2 Overflow
11	0x0014	TIMER1 CAPT	Timer/Counter1 Capture Event
12	0x0016	TIMER1 COMPA	Timer/Counter1 Compare Match A
13	0x0018	TIMER1 COMPB	Timer/Counter1 Compare Match B
14	0x001A	TIMER1 OVF	Timer/Counter1 Overflow
15	0x001C	TIMER0 COMPA	Timer/Counter0 Compare Match A
16	0x001E	TIMER0 COMPB	Timer/Counter0 Compare Match B
17	0x0020	TIMER0 OVF	Timer/Counter0 Overflow
18	0x0022	SPI, STC	SPI Serial Transfer Complete
19	0x0024	USART, RX	USART Rx Complete
20	0x0026	USART, UDRE	USART, Data Register Empty
21	0x0028	USART, TX	USART, Tx Complete
22	0x002A	ADC	ADC Conversion Complete
23	0x002C	EE READY	EEPROM Ready
24	0x002E	ANALOG COMP	Analog Comparator
25	0x0030	TWI	2-wire Serial Interface
26	0x0032	SPM READY	Store Program Memory Ready

No mínimo, **um vetor de interrupção** deve ser sempre programado.  
Este é o vetor de **RESET**.

# RESET do microcontrolador

---

Em geral, o vetor de RESET está nos **últimos endereços de memória**.

Nele se grava o endereço da **primeira instrução** que será executada pelo microcontrolador ao ser iniciado\*.

\* Há diversas formas de se ativar o RESET do microcontrolador (ex.: detecção de energia, erros de software, instrução específica, etc.)



# Tratamento de ISR

---

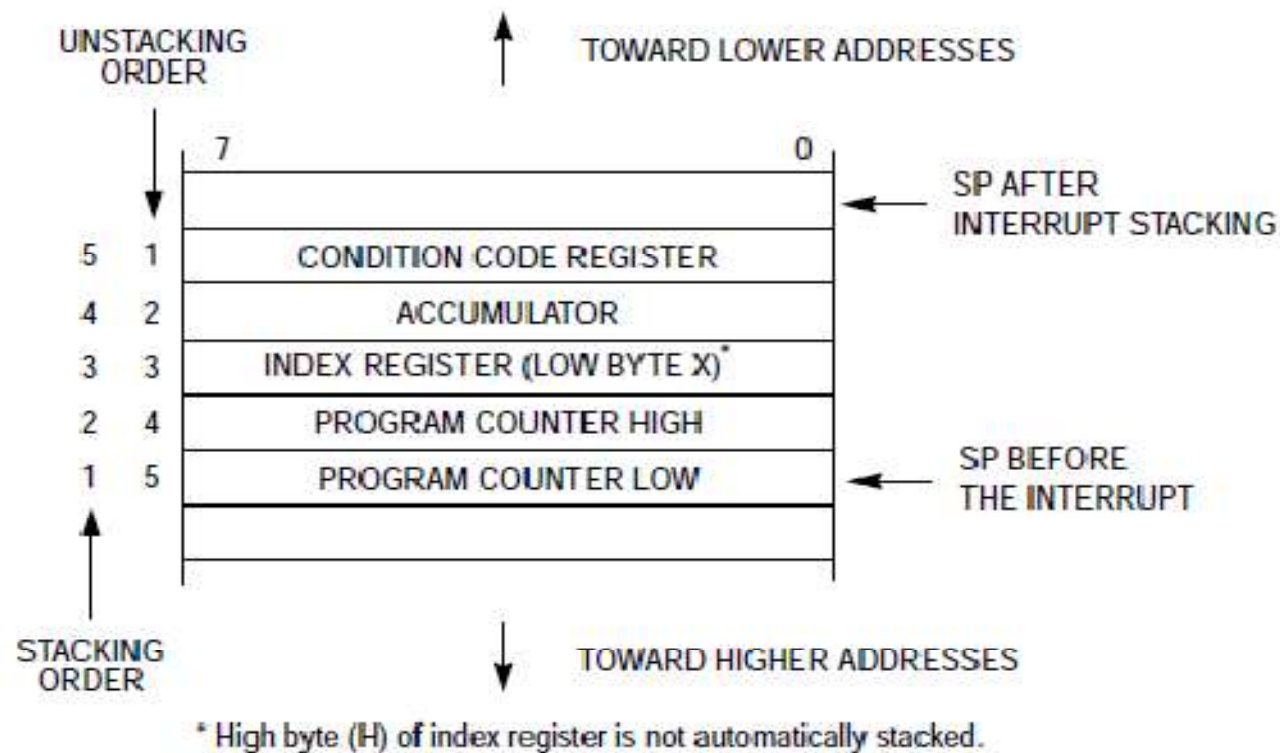
ISR significa sub-rotina de interrupção.

Etapas de execução para tratamento de ISR:

- ocorrência de interrupção com bit I em zero (interrupções habilitadas)
- CPU interrompe a execução do código atual.
- salva seu estado na pilha (memória RAM) – contexto atual.
- coloca bit I para 1 e passa a executar o código de tratamento apontado pelo respectivo vetor de ISR.

# Salvamento de contexto

---



CPU salva registradores PCL, PCH, X, A e CCR

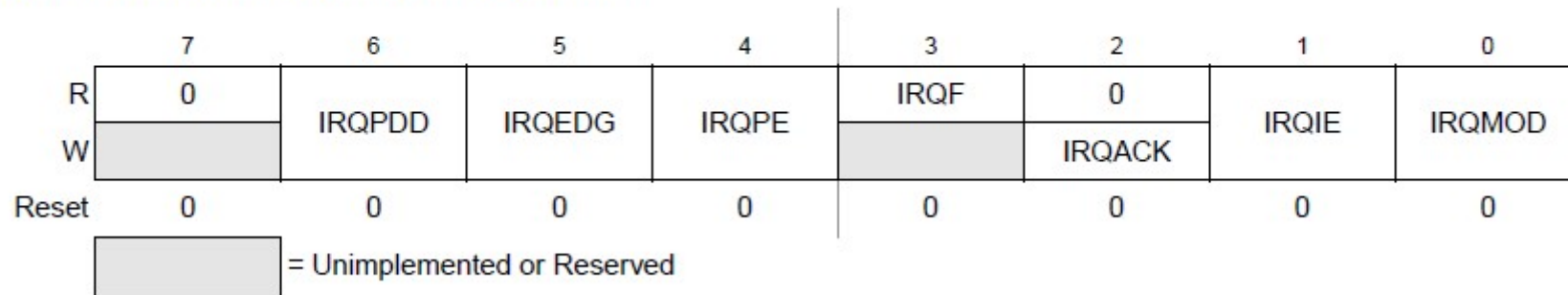
**!!! Exemplo referente a um microcontrolador HCS08 da NXP –**  
**[www.nxp.com/assets/documents/data/en/reference-manuals/HCS08RMV1.pdf](http://www.nxp.com/assets/documents/data/en/reference-manuals/HCS08RMV1.pdf)**

# Exercício

- Interrupção (IRQ)

## Interrupt Pin Request Status and Control Register (IRQSC)

This direct page register includes status and control bits which are used to configure the IRQ function, report status, and acknowledge IRQ events.



Interrupt Request Status and Control Register (IRQSC)

**!!! Exemplo referente a um microcontrolador HCS08 da NXP –**  
**[www.nxp.com/assets/documents/data/en/reference-manuals/HCS08RMV1.pdf](http://www.nxp.com/assets/documents/data/en/reference-manuals/HCS08RMV1.pdf)**

# Tratamento de ISR

---

## SREG – AVR Status Register

The AVR Status Register – SREG – is defined as:

Bit	7	6	5	4	3	2	1	0	
0x3F (0x5F)	I	T	H	S	V	N	Z	C	SREG
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	

### Bit 7 – I: Global Interrupt Enable

Habilita/desabilita as interrupções.

O bit I do SREG é controlado pelas instruções:

CLI – Clear I bit

SEI – Set I bit

RETI – Return from ISR and set I bit

!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# Vetores de interrupção

---

Address	Labels	Code	Comments
0x0000		jmp RESET	; Reset Handler
0x0002		jmp EXT_INT0	; IRQ0 Handler
0x0004		jmp EXT_INT1	; IRQ1 Handler
0x0006		jmp PCINT0	; PCINT0 Handler
0x0008		jmp PCINT1	; PCINT1 Handler
0x000A		jmp PCINT2	; PCINT2 Handler
0x000C		jmp WDT	; Watchdog Timer Handler
0x000E		jmp TIM2_COMPA	; Timer2 Compare A Handler
0x0010		jmp TIM2_COMPB	; Timer2 Compare B Handler
0x0012		jmp TIM2_OVF	; Timer2 Overflow Handler
0x0014		jmp TIM1_CAPT	; Timer1 Capture Handler
0x0016		jmp TIM1_COMPA	; Timer1 Compare A Handler
0x0018		jmp TIM1_COMPB	; Timer1 Compare B Handler
0x001A		jmp TIM1_OVF	; Timer1 Overflow Handler
0x001C		jmp TIM0_COMPA	; Timer0 Compare A Handler
0x001E		jmp TIM0_COMPB	; Timer0 Compare B Handler

!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# ISR de RESET

---

Rotina de tratamento da ISR de RESET:

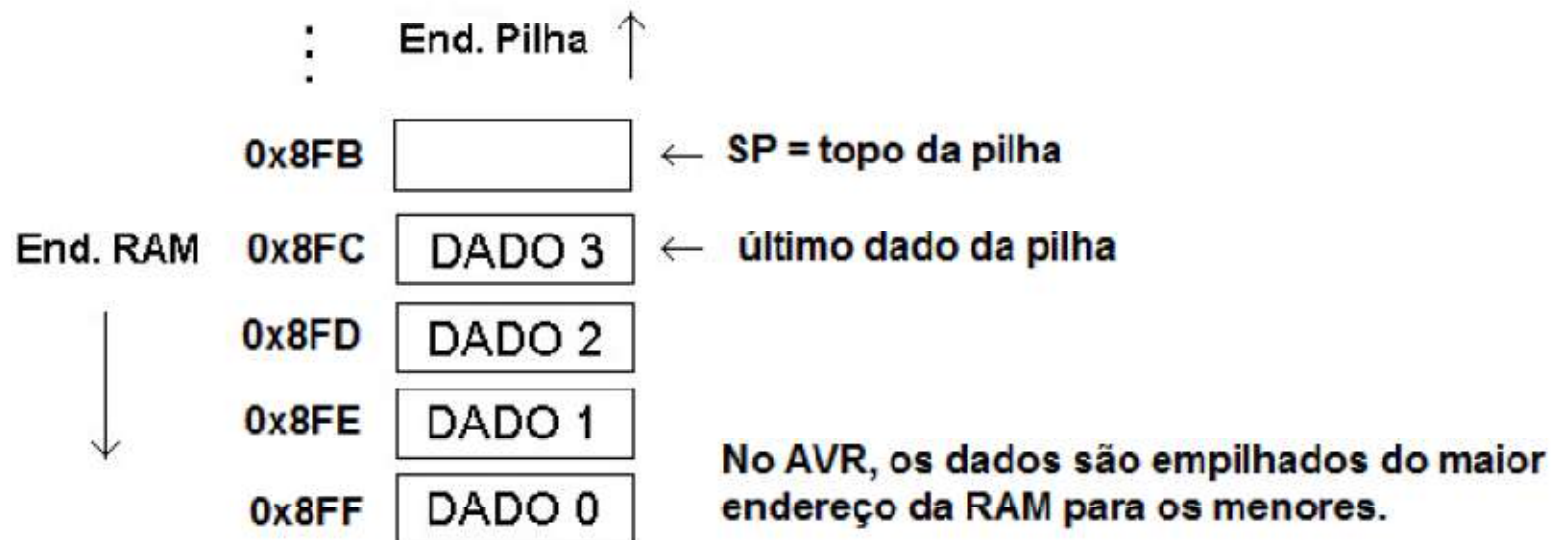
- inicializa o stack pointer
- habilita interrupções

```
0x0033RESET:  ldi    r16, high(RAMEND); Main program start
0x0034        out    SPH,r16          ; Set Stack Pointer to top of RAM
0x0035        ldi    r16, low(RAMEND)
0x0036        out    SPL,r16
0x0037        sei                      ; Enable interrupts
0x0038        <instr>  xxx
```

!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# Pilha (*stack*)

---



Bit	15	14	13	12	11	10	9	8
SPH	-	-	-	-	SP11	SP10	SP9	SP8
SPL	SP7	SP6	SP5	SP4	SP3	SP2	SP1	SP0
Bit	7	6	5	4	3	2	1	0

!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>



!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# Definições de ISRs

```
#define INT0_vect      _VECTOR(1)    /* External Interrupt Request 0 */
#define INT1_vect      _VECTOR(2)    /* External Interrupt Request 1 */
#define PCINT0_vect    _VECTOR(3)    /* Pin Change Interrupt Request 0 */
#define PCINT1_vect    _VECTOR(4)    /* Pin Change Interrupt Request 0 */
#define PCINT2_vect    _VECTOR(5)    /* Pin Change Interrupt Request 1 */
#define WDT_vect       _VECTOR(6)    /* Watchdog Time-out Interrupt */
#define TIMER2_COMPA_vect _VECTOR(7) /* Timer/Counter2 Compare Match A */
#define TIMER2_COMPB_vect _VECTOR(8) /* Timer/Counter2 Compare Match A */
#define TIMER2_OVF_vect _VECTOR(9)   /* Timer/Counter2 Overflow */
#define TIMER1_CAPT_vect _VECTOR(10)  /* Timer/Counter1 Capture Event */
#define TIMER1_COMPA_vect _VECTOR(11) /* Timer/Counter1 Compare Match A */
#define TIMER1_COMPB_vect _VECTOR(12) /* Timer/Counter1 Compare Match B */
#define TIMER1_OVF_vect  _VECTOR(13)  /* Timer/Counter1 Overflow */
#define TIMER0_COMPA_vect _VECTOR(14) /* TimerCounter0 Compare Match A */
#define TIMER0_COMPB_vect _VECTOR(15) /* TimerCounter0 Compare Match B */
#define TIMER0_OVF_vect  _VECTOR(16)  /* Timer/Couner0 Overflow */
#define SPI_STC_vect     _VECTOR(17)  /* SPI Serial Transfer Complete */
#define USART_RX_vect    _VECTOR(18)  /* USART Rx Complete */
#define USART_UDRE_vect  _VECTOR(19)  /* USART, Data Register Empty */
#define USART_TX_vect    _VECTOR(20)  /* USART Tx Complete */
#define ADC_vect         _VECTOR(21)  /* ADC Conversion Complete */
#define EE_READY_vect    _VECTOR(22)  /* EEPROM Ready */
#define ANALOG_COMP_vect _VECTOR(23)  /* Analog Comparator */
#define TWI_vect         _VECTOR(24)  /* Two-wire Serial Interface */
#define SPM_READY_vect   _VECTOR(25)  /* Store Program Memory Read */
```



# Definições de ISRs

---

- As interrupções possuem **diferentes prioridades**: quanto menor é o endereço, maior é a prioridade da interrupção correspondente;
- O **desvio** para uma rotina de interrupção leva pelo menos **4 ciclos de *clock***;
- O **retorno** da interrupção também leva **4 ciclos de *clock***;

!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# Interrupção externa

---

- Interrupção externa (PCINT):

## ► Pin Change Interrupt Control Register

Bit	7	6	5	4	3	2	1	0	
(0x68)	-	-	-	-	-	PCIE2	PCIE1	PCIE0	PCICR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

PCIEx habilita um conjunto de pinos para interrupção:

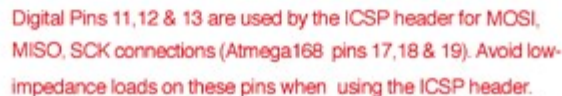
PCIE2 habilita PCINT[23:16]

PCIE1 habilita PCINT[14:8]

PCIE0 habilita PCINT[7:0]

!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

- ### Atmospheric Pollution



# Open

# Interrupção externa

- Interrupção externa (PCINT):

## PCIFR – Pin Change Interrupt Flag Register

Bit	7	6	5	4	3	2	1	0	
0x1B (0x3B)	–	–	–	–	–	PCIF2	PCIF1	PCIF0	PCIFR
Read/Write	R	R	R	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**Bits de *flag* indicam pedidos de interrupção.**

## PCMSK1 – Pin Change Mask Register 1

Bit	7	6	5	4	3	2	1	0	
(0x6C)	–	PCINT14	PCINT13	PCINT12	PCINT11	PCINT10	PCINT9	PCINT8	PCMSK1
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

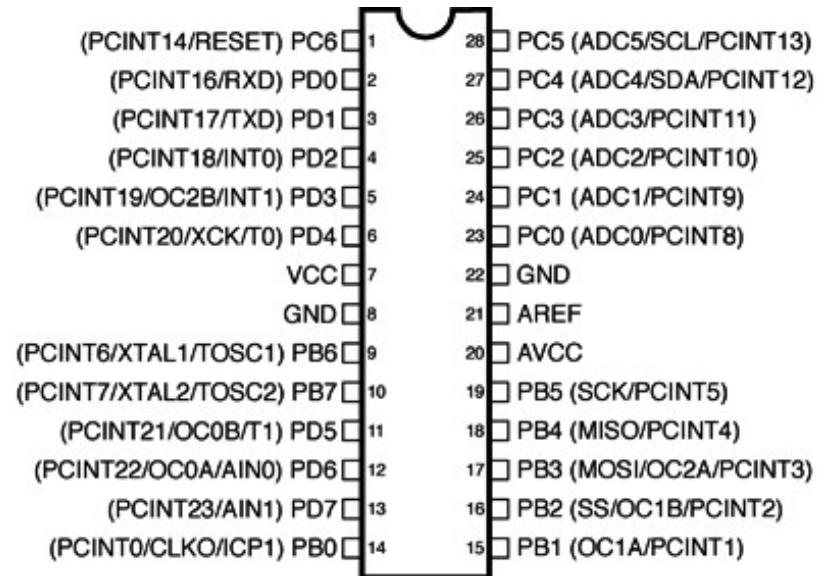
**Bits de *máscara* permitem habilitar pinos individualmente.**

!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# Exercício

---

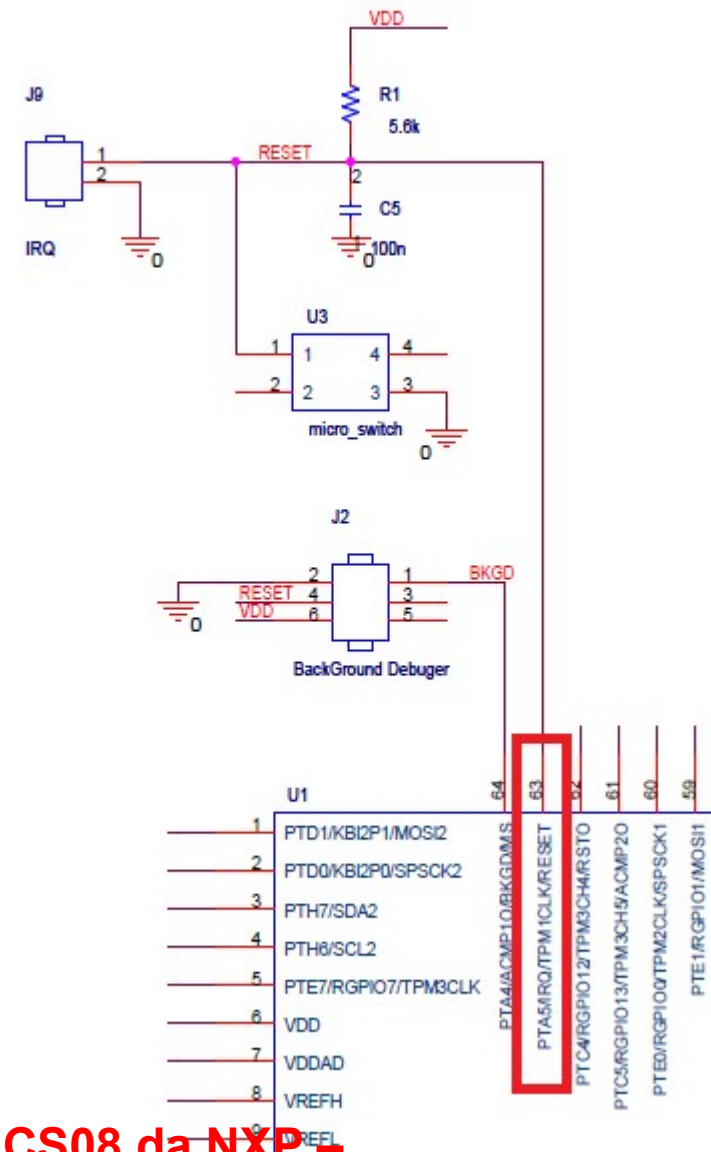
- Interrupção externa (PCINT):
  - Projeto com tratamento da ISR do pino PC6 (PCINT14)



!!! Exemplo referente a um microcontrolador AVR da Microchip/Atmel–  
<http://www.atmel.com/products/microcontrollers/avr/>

# Exercício

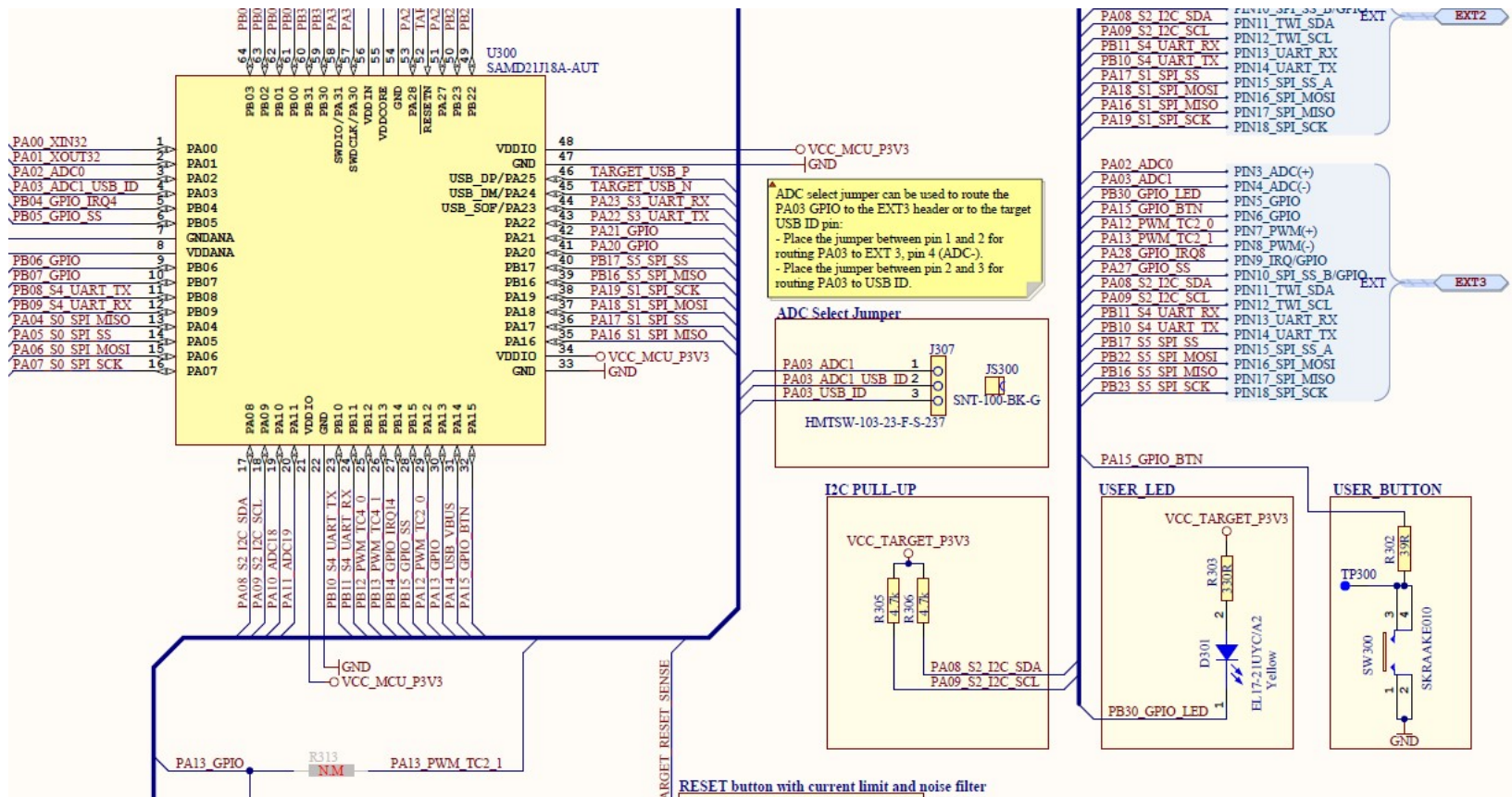
- Interrupção (IRQ)



!!! Exemplo referente a um microcontrolador HCS08 da NXP –

[www.nxp.com/assets/documents/data/en/reference-manuals/HCS08RMV1.pdf](http://www.nxp.com/assets/documents/data/en/reference-manuals/HCS08RMV1.pdf)

- Interrupção (EXTINT)





# Exercício SAM R21

- Interrupção (EXTINT)

