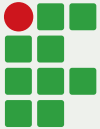
Programando com Interrupções

Interrupção Externa do Atmega328p

Prof. Roberto de Matos

roberto.matos@ifsc.edu.br



INSTITUTO FEDERAL
Santa Catarina
Câmpus São José

Objetivo

Objetivo

- Entender o conceito de interrupção
- *Round Robin* clássico vs. com interrupção
- Entender os mecanismos de interrupção da família ATmega
- Praticar com a interrupção externa do ATmega328p
- Praticar a montagem de circuito e a criação de programas em assembly

[1] Alguns exemplos e imagens dessa apresentação são retirados desse livro.

Objetivo

- Entender o conceito de interrupção
- *Round Robin* clássico vs. com interrupção
- Entender os mecanismos de interrupção da família ATmega
- Praticar com a interrupção externa do ATmega328p
- Praticar a montagem de circuito e a criação de programas em assembly

Referências:

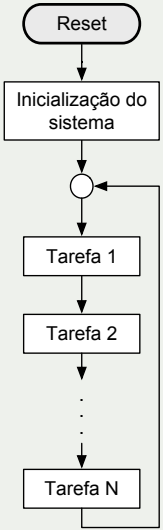
- Datasheet do ATmega328p
- Manual online do montador
- Manual do conjunto de instruções do AVR
- AVR e Arduino: Técnicas de Projeto, 2ª ed.[1] (Leitura recomendada: Capítulo 6)

[1] Alguns exemplos e imagens dessa apresentação são retirados desse livro.

Relembrando ...

Software embarcado básico

- Conhecido como *Round-robin* ou “loopão”
- Simplicidade
- Funcionamento básico:
  - Configura o sistema logo depois do *reset*
  - Executa as tarefas de forma sequencial e repetida

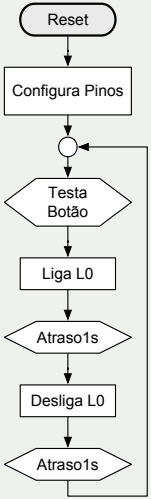


Motivação

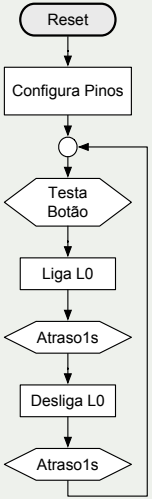
Desafio

Implementar um sistema com um botão (PD2) e dois LEDs, L0 (PB0) e o L1 (PB1). O sistema deve piscar o L0 a cada 2s e quando o botão for pressionado, o L1 deve acender.

Desafio


Reset
Configura Pinos
Testa Botão
Liga L0
Atraso1s
Desliga L0
Atraso1s

Desafio

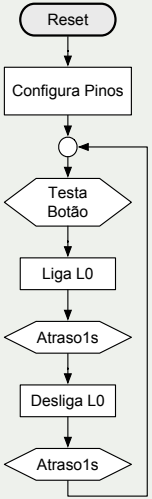



```
1  .INCLUDE <m328pdef.INC>
2
3  .equ BOTAO = PD2
4  .equ L0 = PB0
5  .equ L1 = PB1
6  .def AUX = R16
7
8  setup:
9    LDI AUX,0b00000011
10   OUT DDRB,AUX      ;configura PB3/2 como saída
11   OUT PORTB,AUX     ;desliga os LEDs
12   CBI DDRD, BOTAO   ;configura o PD2 como entrada
13   SBI PORTD, BOTAO ;liga o pull-up do PD2
14
15 main:
16   RCALL testa_botao
17
18   ; Pisca L2
19   SBI  PORTB,L0   ;desliga L0
20
21   LDI r19, 80
22   RCALL delay
23
24   CBI  PORTB,L0   ;liga L0
25
26   LDI r19, 80
27   RCALL delay
28
29   RJMP main
```

```
30
31 ;-------------------------------------------------
32 ; Sub-rotina: Testa BOTAO e liga L1 se pressionado
33 ;-------------------------------------------------
34 testa_botao:
35   SBIC PIND,BOTAO ;botao press. salta a próxima ins.
36   RJMP desliga
37   CBI  PORTB,L1   ;liga L1
38   RJMP fim
39 desliga:
40   SBI  PORTB,L1   ;desliga L1
41 fim:
42   RET
```

Obs.: Rotina *delay* programável foi omitida.

Desafio




```
1  .INCLUDE <m328pdef.INC>
2
3  .equ BOTAO = PD2
4  .equ L0 = PB0
5  .equ L1 = PB1
6  .def AUX = R16
7
8  setup:
9    LDI AUX,0b00000011
10   OUT DDRB,AUX      ;configura PB3/2 como saída
11   OUT PORTB,AUX     ;desliga os LEDs
12   CBI DDRD, BOTAO   ;configura o PD2 como entrada
13   SBI PORTD, BOTAO ;liga o pull-up do PD2
14
15 main:
16   RCALL testa_botao
17
18   ; Pisca L2
19   SBI  PORTB,L0   ;desliga L0
20
21   LDI r19, 80
22   RCALL delay
23
24   CBI  PORTB,L0   ;liga L0
25
26   LDI r19, 80
27   RCALL delay
28
29   RJMP main
```

```
30
31 ;-------------------------------------------------
32 ; Sub-rotina: Testa BOTAO e liga L1 se pressionado
33 ;-------------------------------------------------
34 testa_botao:
35   SBIC PIND,BOTAO ;botao press. salta a próxima ins.
36   RJMP desliga
37   CBI  PORTB,L1   ;liga L1
38   RJMP fim
39 desliga:
40   SBI  PORTB,L1   ;desliga L1
41 fim:
42   RET
```

Obs.: Rotina *delay* programável foi omitida.

Problema

Falta de interatividade

Interrupções

Conceito Geral de Interrupções

- Interrupção é um processo pelo qual um dispositivo externo ou interno pode interromper a execução de uma determinada tarefa do microcontrolador e solicitar a execução de outra.
- A principal vantagem da programação com interrupções é que vários processos podem ser executados em paralelo e que todas as condições decorrentes, a partir de uma pré-configuração, podem ser tratadas no devido tempo.
- Por exemplo, o programa principal pode ser interrompido quando:
  - Um certo tempo configurado passou
  - Uma mensagem chegou por um canal de comunicação
  - Um botão foi pressionado

Conceito Geral de Interrupções

- Interrupção é um processo pelo qual um dispositivo externo ou interno pode interromper a execução de uma determinada tarefa do microcontrolador e solicitar a execução de outra.
- A principal vantagem da programação com interrupções é que vários processos podem ser executados em paralelo e que todas as condições decorrentes, a partir de uma pré-configuração, podem ser tratadas no devido tempo.
- Por exemplo, o programa principal pode ser interrompido quando:
  - Um certo tempo configurado passou
  - Uma mensagem chegou por um canal de comunicação
  - Um botão foi pressionado

Problema:

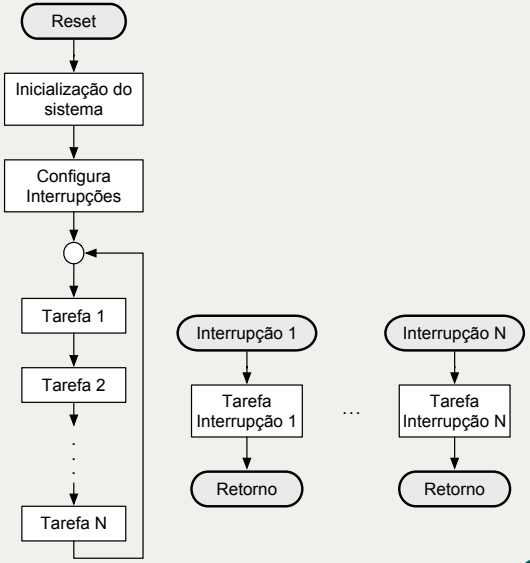
E se duas ou mais interrupções ocorrerem ao mesmo tempo?

Round-Robin com Interrupções

- Eventos urgentes geram interrupções:
  - Tarefas de alta prioridade são tratadas dentro da rotinas de interrupção (ISR).
  - Tarefas de baixa prioridade são tratadas como tarefas *Round-Robin* no *loop* principal.

Round-Robin com Interrupções

- Eventos urgentes geram interrupções:
  - Tarefas de alta prioridade são tratadas dentro da rotinas de interrupção (ISR).
  - Tarefas de baixa prioridade são tratadas como tarefas *Round-Robin* no *loop* principal.



Round-Robin Clássico vs. com Interrupções

- Prós:
  - Evita o uso da CPU com verificações desnecessárias
  - Permite execução **NÃO** bloqueante de tarefas
  - Possibilita o uso do modo de economia de energia de forma mais eficiente
- Contras:
  - Mudança no paradigma linear de execução
  - Imprevisibilidade devido a interrupções assíncronas
  - Aumento da complexidade. Exemplos:
    - Variáveis compartilhadas
    - Condições de corrida

Interrupção na Família AVR ATmega

Características de Interrupção na Família AVR ATmega

- Interrupções vetoradas
- O uso de interrupções aninhadas não é automático
- Prioridade fixa
- Todas as interrupções são mascaráveis
- Habilitação global de interrupções: bit I do SREG

Processo de Interrupção

- **Passo 1:** A CPU completa a instrução em andamento
- **Passo 2:** Desvio para o vetor de interrupção
  - Salva na pilha o endereço da próxima instrução que seria executada (endereço de retorno)
  - Desvia para a posição de memória correspondente à interrupção (vetor)
  - Zera a *flag* da interrupção atendida
  - Desabilita as interrupções globalmente (bit I do SREG)
- **Passo 3:** Executa o código da interrupção
- **Passo 4:** Ao encontrar a instrução `reti` (*RETurn from Interruption*).
  - Carregado no PC o endereço de retorno salvo na pilha
  - Habilita as interrupções globalmente (bit I do SREG)
- **Passo 5:** A execução continua de onde foi interrompida

Processo de Interrupção

- **Passo 1:** A CPU completa a instrução em andamento
- **Passo 2:** Desvio para o vetor de interrupção
  - Salva na pilha o endereço da próxima instrução que seria executada (endereço de retorno)
  - Desvia para a posição de memória correspondente à interrupção (vetor)
  - Zera a *flag* da interrupção atendida
  - Desabilita as interrupções globalmente (bit I do SREG)
- **Passo 3:** Executa o código da interrupção
- **Passo 4:** Ao encontrar a instrução `reti` (*RETurn from Interruption*).
  - Carregado no PC o endereço de retorno salvo na pilha
  - Habilita as interrupções globalmente (bit I do SREG)
- **Passo 5:** A execução continua de onde foi interrompida

É preciso salvar o contexto do SREG e dos registradores utilizados na rotina de interrupção.

Reset e Vetores de Interrupção do ATmega328p

VectorNo.	Program Address[2]	Source	Interrupt Definition
1	0x0000[1]	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/Coutner1 Compare Match B

Vetores de Interrupção do ATmega328p(cont.)

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

Esquema Habilitação das Interrupções

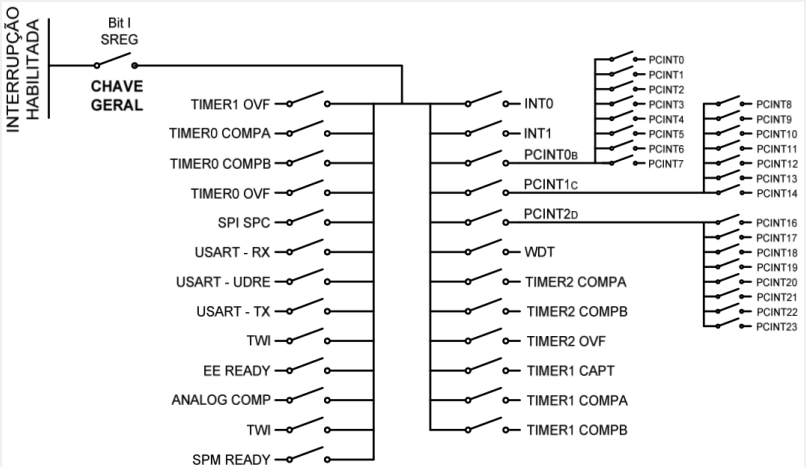


Fig. 6.1 – Chaves de habilitação das interrupções.

Interrupção Externa

Painel de controle - Registradores

Bit	7	6	5	4	3	2	1	0
EICRA	-	-	-	-	ISC11	ISC10	ISC01	ISC00
Lê/Escreve	L	L	L	L	L/E	L/E	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
EIMSK	-	-	-	-	-	-	INT1	INT0
Lê/Escreve	L	L	L	L	L	L	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
EIFR	-	-	-	-	-	-	INTF1	INTF0
Lê/Escreve	L	L	L	L	L	L	L/E	L/E
Valor Inicial	0	0	0	0	0	0	0	0

EICRA – *Extern Interrupt Control Register A*

Bit		7	6	5	4	3	2	1	0
	EICRA	-	-	-	-	ISC11	ISC10	ISC01	ISC00
Lê/Escreve		L	L	L	L	L/E	L/E	L/E	L/E
Valor Inicial		0	0	0	0	0	0	0	0

EICRA – *Extern Interrupt Control Register A*

Tab. 6.2 – Bits de configuração da forma das interrupções nos pinos INT1 e INT0.

ISC11	ISC10	Descrição
0	0	Um nível baixo em INT1 gera um pedido de interrupção.
0	1	Qualquer mudança lógica em INT1 gera um pedido de interrupção.
1	0	Uma borda de decida em INT1 gera um pedido de interrupção.
1	1	Uma borda de subida em INT1 gera um pedido de interrupção.

ISC01	ISC00	Descrição
0	0	Um nível baixo em INT0 gera um pedido de interrupção.
0	1	Qualquer mudança lógica em INT0 gera um pedido de interrupção.
1	0	Uma borda de decida em INT0 gera um pedido de interrupção.
1	1	Uma borda de subida em INT0 gera um pedido de interrupção.

EIMSK – *External Interrupt Mask Register*

Bit		7	6	5	4	3	2	1	0
	EIMSK	-	-	-	-	-	-	INT1	INT0
Lê/Escreve		L	L	L	L	L	L	L/E	L/E
Valor Inicial		0	0	0	0	0	0	0	0

EIFR - *External Interrupt Flag Register*

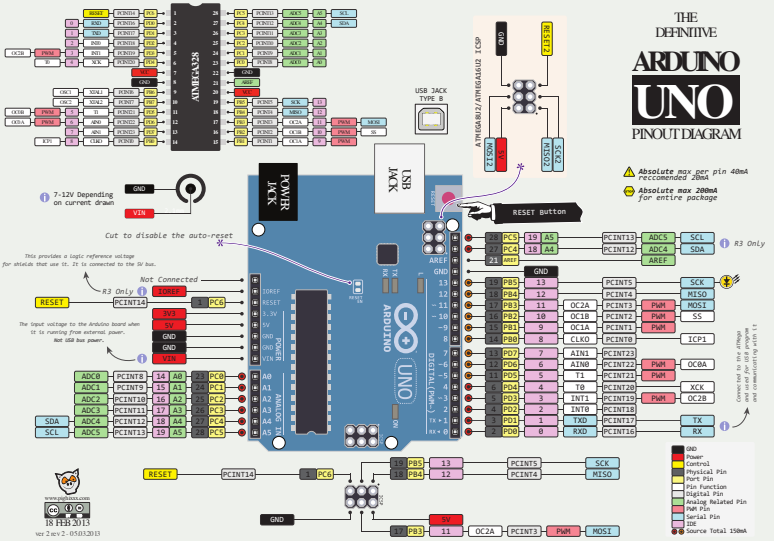
Bit		7	6	5	4	3	2	1	0
	EIFR	-	-	-	-	-	-	INTF1	INTF0
Lê/Escreve		L	L	L	L	L	L	L/E	L/E
Valor Inicial		0	0	0	0	0	0	0	0

Exemplo

Exemplo - Fluxograma

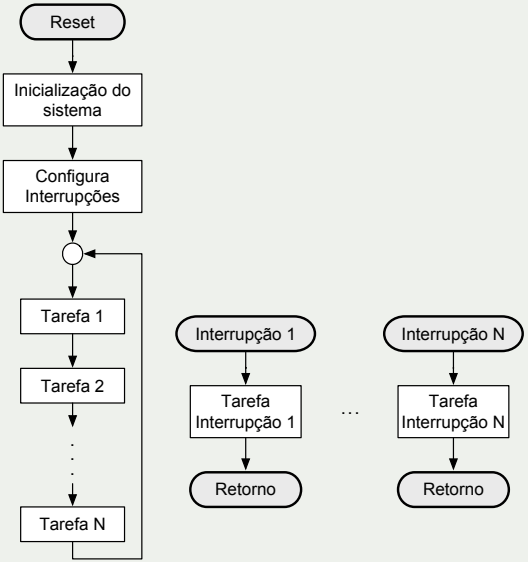
Implementar um sistema com um botão (PD2) e dois LEDs, L0 (PB0) e o L1 (PB1). O sistema deve piscar o L0 a cada 2s e quando o botão for pressionado, o L1 deve acender.

Exemplo - Fluxograma



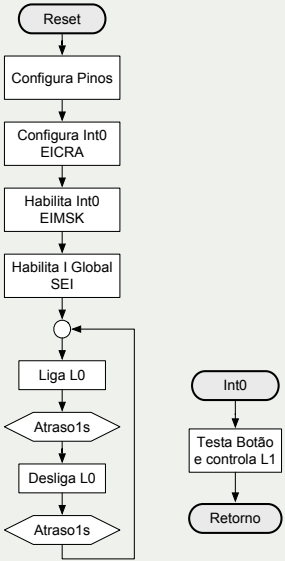
Exemplo - Fluxograma

Implementar um sistema com um botão (PD2) e dois LEDs, L0 (PB0) e o L1 (PB1). O sistema deve piscar o L0 a cada 2s e quando o botão for pressionado, o L1 deve acender.



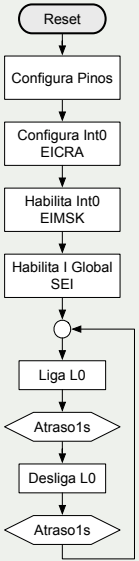
Exemplo - Fluxograma

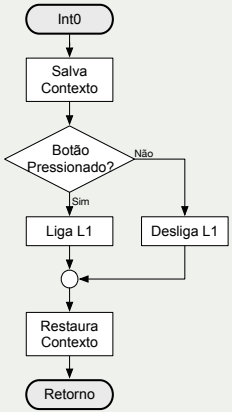
Implementar um sistema com um botão (PD2) e dois LEDs, L0 (PB0) e o L1 (PB1). O sistema deve piscar o L0 a cada 2s e quando o botão for pressionado, o L1 deve acender.



Exemplo - Fluxograma

Implementar um sistema com um botão (PD2) e dois LEDs, L0 (PB0) e o L1 (PB1). O sistema deve piscar o L0 a cada 2s e quando o botão for pressionado, o L1 deve acender.





Exemplo - Assembly (Obs.: Rotina *delay* programável foi omitida)

```
1  .INCLUDE <m328pdef.INC>
2
3  ;DEFINIÇÕES
4  .equ BOTAO = PD2
5  .equ L0 = PB0
6  .equ L1 = PB1
7  .def AUX = R16
8
9  .ORG 0x0000          ; Reset vector
10   RJMP setup
11
12 .ORG 0x0002         ; Vetor (endereço na Flash) da INT0
13   RJMP isr_int0
14
15 .ORG 0x0034         ; primeira end. livre depois dos vetores
16 setup:
17   LDI AUX,0x03      ; 0b00000011
18   OUT DDRB,AUX      ; configura PB3/2 como saída
19   OUT PORTB,AUX     ; desliga os LEDs
20   CBI DDRD, BOTAO   ; configura o PD2 como entrada
21   SBI PORTD, BOTAO  ; liga o pull-up do PD2
22
23   LDI AUX, 0x01     ;
24   STS EICRA, AUX    ; config. INT0 sensível a borda
25   SBI EIMSK, INT0   ; habilita a INT0
26
27   SEI               ; habilita a interrupção global ...
28                     ; ... (bit I do SREG)
```

```
29 main:
30   SBI  PORTB,L0     ; desliga L0
31   LDI r19, 80
32   RCALL delay       ; delay 1s
33   CBI  PORTB,L0     ; liga L0
34   LDI r19, 80
35   RCALL delay       ; delay 1s
36   RJMP main
37
38 ;-----------------------------------------------
39 ; Rotina de Interrupção (ISR) da INT0
40 ;-----------------------------------------------
41 isr_int0:
42   PUSH R16          ; Salva o contexto (SREG)
43   IN R16, SREG
44   PUSH R16
45
46   SBIC PIND,BOTAO ; botao press. salta a próxima inst.
47   RJMP desliga
48   CBI  PORTB,L1   ; liga L1
49   RJMP fim
50 desliga:
51   SBI  PORTB,L1   ; desliga L1
52
53 fim:
54   POP R16           ; Restaura o contexto (SREG)
55   OUT SREG,R16
56   POP R16
57   RETI              ; retorna da interrupcao
```

Experimentos

Experimentos no MPLAB IDE

- Testar e gravar todos os trechos de códigos apresentados.
- Modifique o circuito e o código exemplo para funcionar com o botão no PD3 (int1).

Fim!