Desenvolvimento de Sistemas Embarcados em Tempo Real

Prof. Hermano Cabral

Departmento de Eletrônica e Sistemas — UFPE

21 de agosto de 2018

Plano de Aula

Tema central

- Arquitetura AVR — interrupções

Plano de Aula

Tema central

- Arquitetura AVR — interrupções

Objetivos

- Conhecer como a arquitetura AVR lida com interrupções
- Conhecer as diferentes fontes de interrupções
- Desenvolver rotinas de tratamento de interrupções
- Saber habilitar e desabilitar as diferentes interrupções

Introdução

- Um microcontrolador tem 2 maneiras de descobrir se há algum evento a ser tratado:
  - Polling

Introdução

- Um microcontrolador tem 2 maneiras de descobrir se há algum evento a ser tratado:
  - Polling
  - Interrupção

Introdução - Polling

- Polling consiste de um loop cuja condição de saída é a ocorrência de um evento.

Introdução - Polling

- Polling consiste de um loop cuja condição de saída é a ocorrência de um evento.
- Polling é simples de implementar pois está no fluxo sequencial do programa.

Introdução - Polling

- Polling consiste de um loop cuja condição de saída é a ocorrência de um evento.
- Polling é simples de implementar pois está no fluxo sequencial do programa.
- A desvantagem é que, se o evento não é frequente, o microcontrolador perde muito tempo fazendo o polling.

Introdução - Polling

- Polling consiste de um loop cuja condição de saída é a ocorrência de um evento.
- Polling é simples de implementar pois está no fluxo sequencial do programa.
- A desvantagem é que, se o evento não é frequente, o microcontrolador perde muito tempo fazendo o polling.
- Se colocarmos instruções entre os momentos de polling, podemos criar jitter.

Introdução - interrupções

- Interrupções são um mecanismo que permite o MCU ser interrompido quando o evento ocorre.

Introdução - interrupções

- Interrupções são um mecanismo que permite o MCU ser interrompido quando o evento ocorre.
- A vantagem é que o MCU pode fazer outras coisas enquanto o evento não ocorre.

Introdução - interrupções

- Interrupções são um mecanismo que permite o MCU ser interrompido quando o evento ocorre.
- A vantagem é que o MCU pode fazer outras coisas enquanto o evento não ocorre.
- A desvantagem é que a interrupção irá executar concorrentemente com o programa.

Introdução - interrupções

- Interrupções são um mecanismo que permite o MCU ser interrompido quando o evento ocorre.
- A vantagem é que o MCU pode fazer outras coisas enquanto o evento não ocorre.
- A desvantagem é que a interrupção irá executar concorrentemente com o programa.
- Outra desvantagem é que leva um certo tempo para a MCU parar o que estava fazendo e tratar a interrupção.

Arquitetura AVR

- A arquitetura AVR possui diferentes fontes de interrupções.

Arquitetura AVR

- A arquitetura AVR possui diferentes fontes de interrupções.
- Algumas vezes, um determinado periférico pode disparar mais de um tipo de interrupção.

Arquitetura AVR

- A arquitetura AVR possui diferentes fontes de interrupções.
- Algumas vezes, um determinado periférico pode disparar mais de um tipo de interrupção.
- Por exemplo, o temporizador 0 tem 3 interrupções distintas que podem ser ativadas.

Arquitetura AVR

- A arquitetura AVR possui diferentes fontes de interrupções.
- Algumas vezes, um determinado periférico pode disparar mais de um tipo de interrupção.
- Por exemplo, o temporizador 0 tem 3 interrupções distintas que podem ser ativadas.
- Ao todo, o processador ATMega328p possui 26 tipos diferentes de interrupções, enquanto o ATMega32u4 possui 37 tipos.

Interrupções

Vector No	Program Address(2)	Source	Interrupts 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 0
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

Tratamento de interrupções

- Cada interrupção possui uma posição fixa na memória contendo o endereço de uma função a ser chamada para tratá-la.

Tratamento de interrupções

- Definimos uma rotina de tratamento de interrupção em C com a construção `ISR(INT0_vect)`.

Tratamento de interrupções

- Definimos uma rotina de tratamento de interrupção em C com a construção `ISR(INT0_vect)`.
- Isto se encarrega de preencher a tabela de vetores com o endereço correto da função.

Tratamento de interrupções

- Além disso, cada interrupção também tem um bit de habilitação do seu tratamento em um registrador apropriado de controle.

Tratamento de interrupções

- Além disso, cada interrupção também tem um bit de habilitação do seu tratamento em um registrador apropriado de controle.
- Assim, a interrupção só é tratada se o bit de habilitação global de interrupções e seu bit de habilitação específico tiverem o valor 1.

Tratamento de interrupções

- Além disso, cada interrupção também tem um bit de habilitação do seu tratamento em um registrador apropriado de controle.
- Assim, a interrupção só é tratada se o bit de habilitação global de interrupções e seu bit de habilitação específico tiverem o valor 1.
- Caso contrário, a interrupção é ignorada.

Tratamento de interrupções

- A maioria das fontes de interrupção possuem também um bit de flag que é setado pelo evento que gerou a interrupção.

Tratamento de interrupções

- A maioria das fontes de interrupção possuem também um bit de flag que é setado pelo evento que gerou a interrupção.
- Este bit é setado independente do estado dos bits de habilitação de interrupção.

Tratamento de interrupções

- A maioria das fontes de interrupção possuem também um bit de flag que é setado pelo evento que gerou a interrupção.
- Este bit é setado independente do estado dos bits de habilitação de interrupção.
- Eles podem ser usados para verificar se uma interrupção aconteceria mesmo se ela não estiver habilitada.

Tratamento de interrupções

- A maioria das fontes de interrupção possuem também um bit de flag que é setado pelo evento que gerou a interrupção.
- Este bit é setado independente do estado dos bits de habilitação de interrupção.
- Eles podem ser usados para verificar se uma interrupção aconteceria mesmo se ela não estiver habilitada.
- Este tipo de tratamento de eventos recebe o nome de *polling*.

Tratamento de interrupções

- A flag também serve para "lembrar" da interrupção caso esta tenha ocorrido enquanto estava desabilitada.

Tratamento de interrupções

- A flag também serve para "lembrar" da interrupção caso esta tenha ocorrido enquanto estava desabilitada.
- Quando se reabilitar a interrupção, ela será lançada.

Tratamento de interrupções

- A flag também serve para "lembrar" da interrupção caso esta tenha ocorrido enquanto estava desabilitada.
- Quando se reabilitar a interrupção, ela será lançada.
- Por essa razão, é importante garantir que a flag seja resetada antes de voltar da interrupção.

Tratamento de interrupções

- A flag também serve para "lembrar" da interrupção caso esta tenha ocorrido enquanto estava desabilitada.
- Quando se reabilitar a interrupção, ela será lançada.
- Por essa razão, é importante garantir que a flag seja resetada antes de voltar da interrupção.
- Para a maioria das interrupções deste caso, a própria CPU reseta a flag automaticamente ao chamar a rotina da interrupção.

The External Interrupt Control Register A contains control bits for interrupt sense control.

Name: EICRA
Offset: 0x69
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					ISC11	ISC10	ISC01	ISC00
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Tratamento de interrupções — exemplo

- Tomemos o caso da interrupção externa INT0.

The External Interrupt Control Register A contains control bits for interrupt sense control.

Name: EICRA
Offset: 0x69
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
					ISC11	ISC10	ISC01	ISC00
Access					R/W	R/W	R/W	R/W
Reset					0	0	0	0

Tratamento de interrupções — exemplo

- Tomemos o caso da interrupção externa INT0.
- O registrador de controle é o EICRA, mostrado acima, que contém 2 bits de controle do INT0.

Value	Description
00	The low level of INT0 generates an interrupt request.
01	Any logical change on INT0 generates an interrupt request.
10	The falling edge of INT0 generates an interrupt request.
11	The rising edge of INT0 generates an interrupt request.

Tratamento de interrupções — exemplo

- Os valores destes bits dependem de qual evento queremos que gere a interrupção.

Value	Description
00	The low level of INT0 generates an interrupt request.
01	Any logical change on INT0 generates an interrupt request.
10	The falling edge of INT0 generates an interrupt request.
11	The rising edge of INT0 generates an interrupt request.

Tratamento de interrupções — exemplo

- Os valores destes bits dependem de qual evento queremos que gere a interrupção.
- Note que se o modo for de ativação no nível baixo, o sistema fica gerando interrupções enquanto o nível no pino estiver baixo.

Value	Description
00	The low level of INT0 generates an interrupt request.
01	Any logical change on INT0 generates an interrupt request.
10	The falling edge of INT0 generates an interrupt request.
11	The rising edge of INT0 generates an interrupt request.

Tratamento de interrupções — exemplo

- Os valores destes bits dependem de qual evento queremos que gere a interrupção.
- Note que se o modo for de ativação no nível baixo, o sistema fica gerando interrupções enquanto o nível no pino estiver baixo.
- Deve-se observar que as interrupções serão geradas mesmo que o pino esteja configurado como de saída.

Name: EIMSK
Offset: 0x3D
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x1D

Bit	7	6	5	4	3	2	1	0
							INT1	INT0
Access							R/W	R/W
Reset							0	0

Tratamento de interrupções — exemplo

- O registrador EIMSK contém o bit de habilitação específica para o pino INT0.

Name: EIFR
Offset: 0x3C
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x1C

Bit	7	6	5	4	3	2	1	0
							INTF1	INTF0
Access							R/W	R/W
Reset							0	0

Tratamento de interrupções — exemplo

- O registrador EIFR contém a flag relacionada ao evento de INT0.

Name: EIFR
Offset: 0x3C
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x1C

Bit	7	6	5	4	3	2	1	0
							INTF1	INTF0
Access							R/W	R/W
Reset							0	0

Tratamento de interrupções — exemplo

- O registrador EIFR contém a flag relacionada ao evento de INT0.
- Esta flag é resetada quando a rotina de interrupção é executada ou quando se escreve o valor 1 na flag.

Tratamento de interrupções — PCINT

- Outro tipo de interrupção similar ao da interrupção externa INT0 é o do PCINT.

Tratamento de interrupções — PCINT

- Outro tipo de interrupção similar ao da interrupção externa INT0 é o do PCINT.
- Neste caso, a quantidade de pinos disponíveis para interrupção é bem maior.

Tratamento de interrupções — PCINT

- Outro tipo de interrupção similar ao da interrupção externa INT0 é o do PCINT.
- Neste caso, a quantidade de pinos disponíveis para interrupção é bem maior.
- Entretanto, os pinos são divididos em 3 grupos, cada grupo com sua interrupção.

Tratamento de interrupções — PCINT

- Outro tipo de interrupção similar ao da interrupção externa INT0 é o do PCINT.
- Neste caso, a quantidade de pinos disponíveis para interrupção é bem maior.
- Entretanto, os pinos são divididos em 3 grupos, cada grupo com sua interrupção.
- Se qualquer pino de um grupo mudar de nível lógico, a interrupção daquele grupo será ativada.

Tratamento de interrupções — PCINT

- Outro tipo de interrupção similar ao da interrupção externa INT0 é o do PCINT.
- Neste caso, a quantidade de pinos disponíveis para interrupção é bem maior.
- Entretanto, os pinos são divididos em 3 grupos, cada grupo com sua interrupção.
- Se qualquer pino de um grupo mudar de nível lógico, a interrupção daquele grupo será ativada.
- Um registrador armazena o pino que causou a interrupção.

Name: PCICR
Offset: 0x68
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
						PCIE2	PCIE1	PCIE0
Access						R/W	R/W	R/W
Reset						0	0	0

Tratamento de interrupções — PCINT

- O registrador de controle é o PCICR, que neste caso controla a habilitação da interrupção de cada grupo de pinos.

Name: PCIFR
Offset: 0x3B
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x1B

Bit	7	6	5	4	3	2	1	0
						PCIF2	PCIF1	PCIF0
Access						R/W	R/W	R/W
Reset						0	0	0

Tratamento de interrupções — PCINT

- O registrador de flags é o PCIFR, com uma flag para cada grupo.

Name: PCIFR
Offset: 0x3B
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x1B

Bit	7	6	5	4	3	2	1	0
						PCIF2	PCIF1	PCIF0
Access						R/W	R/W	R/W
Reset						0	0	0

Tratamento de interrupções — PCINT

- O registrador de flags é o PCIFR, com uma flag para cada grupo.
- Esta flag também é setada na mudança de nível lógico de um dos pinos do grupo.

Name: PCIFR
Offset: 0x3B
Reset: 0x00
Property: When addressing as I/O Register: address offset is 0x1B

Bit	7	6	5	4	3	2	1	0
						PCIF2	PCIF1	PCIF0
Access						R/W	R/W	R/W
Reset						0	0	0

Tratamento de interrupções — PCINT

- O registrador de flags é o PCIFR, com uma flag para cada grupo.
- Esta flag também é setada na mudança de nível lógico de um dos pinos do grupo.
- A flag é resetada na chamada da rotina de tratamento da interrupção, ou ao se escrever o bit 1 na flag.

Name: PCMSK0
Offset: 0x6B
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Tratamento de interrupções — PCINT

- Podemos habilitar a interrupção no caso de mudança de nível lógico de um pino setando o bit correspondente do registrador PCMSK.

Name: PCMSK0
Offset: 0x6B
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Tratamento de interrupções — PCINT

- Podemos habilitar a interrupção no caso de mudança de nível lógico de um pino setando o bit correspondente do registrador PCMSK.
- Observe que, no caso de PCINT, a interrupção ocorrerá apenas se as seguintes 3 condições ocorrerem:

Name: PCMSK0
Offset: 0x6B
Reset: 0x00
Property: -

Bit	7	6	5	4	3	2	1	0
	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0
Access	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Reset	0	0	0	0	0	0	0	0

Tratamento de interrupções — PCINT

- Podemos habilitar a interrupção no caso de mudança de nível lógico de um pino setando o bit correspondente do registrador PCMSK.
- Observe que, no caso de PCINT, a interrupção ocorrerá apenas se as seguintes 3 condições ocorrerem:
  - A flag global de habilitação estiver setada
  - A flag do grupo de pinos de interesse no registrador PCICR estiver setada
  - A flag do pino individual no registrador PCMSK estiver setada

Tratamento de interrupções — PCINT

- Como a interrupção do PCINT de um grupo de bits é acionada por qualquer bit do grupo, não há informação sobre que bit mudou.

Tratamento de interrupções — PCINT

- Como a interrupção do PCINT de um grupo de bits é acionada por qualquer bit do grupo, não há informação sobre que bit mudou.
- Se quisermos ter esta informação, devemos salvar o estado dos bits do grupo na rotina de interrupção.

Tipos de interrupções

- As 26 interrupções do ATMega328p são divididas entre os vários periféricos e estão mostradas nas tabelas a seguir.

Interrupções

Vector No	Program Address(2)	Source	Interrupts 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 0
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/Coutner2 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

Tipos de interrupção

Vector No	Program Address(2)	Source	Interrupts definition
14	0x001A	TIMER1_OVF	Timer/Counter1 Overflow
15	0x001C	TIMER0_COMPA	Timer/Counter0 Compare Match A
16	0x001E	TIMER0_COMPB	Timer/Coutner0 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 (I^2C)
26	0x0032	SPM READY	Store Program Memory Ready

Tipos de interrupção

Tratamento de interrupções

- Existem 2 regras muito importantes para tratamento de interrupções:

Tratamento de interrupções

- Existem 2 regras muito importantes para tratamento de interrupções:
  - Regra do bem-educado

Tratamento de interrupções

- Existem 2 regras muito importantes para tratamento de interrupções:
  - Regra do bem-educado
  - Regra do lacônico

Regra do bem-educado

- A rotina de interrupção deve deixar a CPU do microcontrolador exatamente como encontrou.

Regra do bem-educado

- A rotina de interrupção deve deixar a CPU do microcontrolador exatamente como encontrou.
- Isto, em geral, é garantido pelo compilador C.

Regra do lacônico

- A rotina de interrupção deve realizar apenas o essencial.

Regra do lacônico

- A rotina de interrupção deve realizar apenas o essencial.
- Se uma tarefa é demorada, a rotina deve fazer o mínimo possível e repassar a maior parte possível para uma tarefa de menor prioridade.