


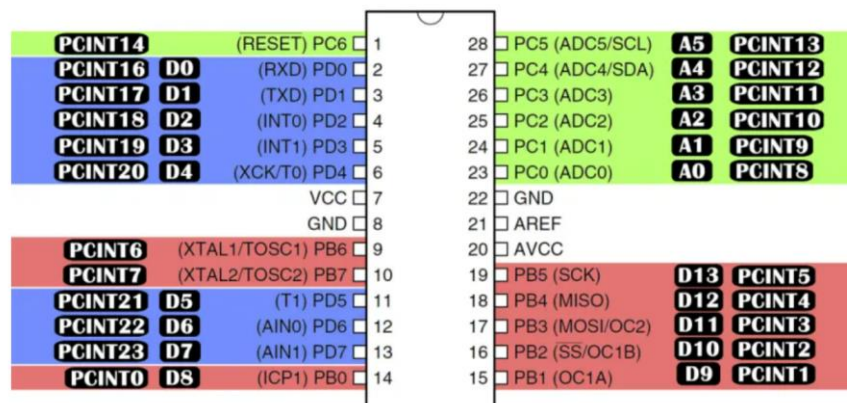
<div> <i>Instituto Nacional de Telecomunicações</i></div>	RELATÓRIO 7	Data:    /    /	
	Disciplina: E209		
	Professor: Yvo Marcelo Chiaradia Masselli		
	Monitores: Diego Coutinho, Lucas Lares, Maria Luiza e Thalita Domingos		
Conteúdo: Microcontrolador ATmega328p			
Tema: Interrupção Externa com PCINT			
Nome:		Matrícula:	Curso:

### OBJETIVOS:

- Utilizar ferramentas de simulação para desenvolver programas para o ATmega328p.
- Desenvolver um programa que faça uso da interrupção externa com baixa prioridade.
- Utilizar as entradas e saídas do ATmega328p com circuitos de aplicação.

### Parte Teórica

**Interrupção com PCINT:** É um recurso no qual o ciclo de execução natural do programa é paralisado para executar um bloco específico. A principal diferença entre o INT e PCINT, é que o PCINT está em quase todos os pinos, e não é possível configurar qual mudança de nível lógico irá interromper, será sempre “qualquer mudança”.



Seu funcionamento não é individual, mas sim em grupos, definidos como grupos de interrupção. Ou seja, ao gerar uma interrupção em um dos pinos de um PORTAL, o serviço da rotina de interrupção associado ao PORTAL é executado, **para diferenciar entre os pinos, é necessário fazer uma leitura do PORTAL** logo que o programa entra na rotina de interrupção, pois o microcontrolador não é capaz de tratar disso por conta própria.

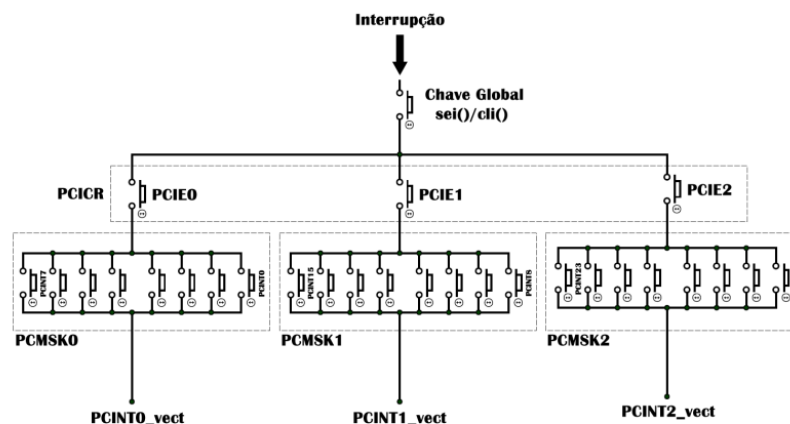


Figura 2. Exemplo que representa o funcionamento do PCINT

No caso do PCINT, existem dois registradores que são configurados para que seja possível gerar uma interrupção. Estes são:

### PCICR – Pin Change Interrupt Control Register

Este registrador é responsável por **habilitar a interrupção** em um determinado **PORT** quando o respectivo **bit PCIE<sub>x</sub>** for alterado para 1.

#### PCICR – 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	

### PCMSK – Pin Change Mask Register

Este registrador é responsável por habilitar a interrupção de um pino em um determinado PORT. Logo, existem 3 registradores desse tipo **PCMSK0**, **PCMSK1** e **PCMSK2** referentes aos **PORTS B, C e D** respectivamente.

#### PCMSK0 – Pin Change Mask Register 0

Bit	7	6	5	4	3	2	1	0	
(0x6B)	PCINT7	PCINT6	PCINT5	PCINT4	PCINT3	PCINT2	PCINT1	PCINT0	PCMSK0
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	

#### 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	

#### PCMSK2 – Pin Change Mask Register 2

Bit	7	6	5	4	3	2	1	0	
(0x6D)	PCINT23	PCINT22	PCINT21	PCINT20	PCINT19	PCINT18	PCINT17	PCINT16	PCMSK2
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	

Para fazer uso de qualquer interrupção no ATmega328p é necessário **habilitar o bit de interrupção global**. A linha de código utilizada para isso é: **sei();**. Essa linha acessa o registrador **SREG** e seta o bit responsável por habilitar o periférico de interrupções como 1, habilitando todas as interrupções no MCU.

Para diferenciar os grupos de interrupção configurados em **PCMSK0**, **PCMSK1** e **PCMSK2**, são utilizados os vetores **PCINT0\_vect**, **PCINT1\_vect** e **PCINT2\_vect** respectivamente.

### Exercícios:

- 1) Escreva um programa que use a interrupção externa **PCINT** como canais de interrupção. O programa deverá piscar um LED no pino 5 (PD5) a cada 0,250 segundos no super loop e a cada vez que acontecer **um pedido de interrupção**, um LED no pino 7 (PD7) acende por 1 segundo quando o PCINT0 for acionado, outro LED no pino 6 (PD6) acende por 0.5 segundos quando o PCINT2 for acionado, e outro LED no pino 15 (PB1) acende por 2 segundos quando o PCINT20 for acionado.

```
// Na interrupção PCINT0
ISR(PCINT0_vect)
{
    /* CÓDIGO QUE VAI RODAR
    NA INTERRUPÇÃO DO PCINT0 */
}

// Na interrupção PCINT1
ISR(PCINT1_vect)
{
    /* CÓDIGO QUE VAI RODAR
    NA INTERRUPÇÃO DO PCINT1 */
}

// Na interrupção PCINT2
ISR(PCINT2_vect)
{
    /* CÓDIGO QUE VAI RODAR
    NA INTERRUPÇÃO DO PCINT2 */
}

int main(void)
{
    // Habilita interrupção no grupo PCINT2
    PCICR |= 0b00000100;

    // Habilita PCINT no pino PD4 -> PCINT20
    PCMSK2 |= 0b00010000;

    // Habilita Interrupções globais
    sei();

    for (;;)
    {
        /* CÓDIGO QUE VAI RODAR
        SEMPRE / SUPER-LOOP */
    }
}
```