 Instituto Nacional de Telecomunicações	RELATÓRIO 2 - Teoria		Data: / /
	Disciplina: E209		
	Prof: João Pedro Magalhães de Paula Paiva		
Conteúdo: Microcontroladores AVR			
Tema: Interrupções			
Nome:		Matrícula:	Curso:

OBJETIVOS:

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

Parte Teórica

Interrupção

É um recurso no qual **o ciclo de execução natural do programa é paralisado** para **executar um bloco específico**. Quando uma interrupção é chamada, caso a mesma esteja habilitada, o microcontrolador salta para um endereço padrão da memória de programa que contem tal bloco específico.

O objetivo desse relatório é estudar a interrupção externa. O ATmega328p permite que a interrupção externa seja disparada por um dos seguintes eventos: presença de um **nível lógico baixo**, **transição de subida**, **transição de descida** ou ainda **transições de subida e descida** do nível lógico presente nos pinos **PD2(INT0)** ou **PD3(INT1)**. Para configurar o evento que vai disparar a interrupção externa, utiliza-se o registro **EICRA**. A figura abaixo mostra os bits desse registro:

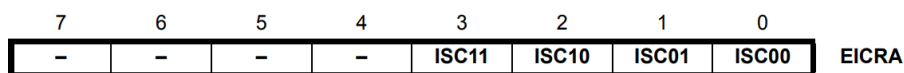


Figura 1 – Registro EICRA

A tabela abaixo apresenta as configurações para selecionar o evento que irá disparar a interrupção.

ISCx1	ISCx0	Descrição
0	0	Presença de nível lógico baixo em INTx gera interrupção.
0	1	Transições de subida e descida do nível lógico presente em INTx geram interrupção.
1	0	Transição de descida do nível lógico presente em INTx gera interrupção.
1	1	Transição de subida do nível lógico presente em INTx gera interrupção.

Tabela 1 – Configuração do registro EICRA

Outro registro que é necessário para utilizar a interrupção externa, é o registro **EIMSK**. Esse registro é responsável por habilitar a interrupção externa desejada. A figura abaixo mostra os bits desse registro:

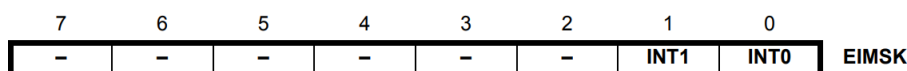


Figura 2 – Registro EIMSK

O bit **INT0** habilita a **interrupção externa 0 (PD2)** e o bit **INT1** habilita a **interrupção externa 1 (PD3)**.

Caso seja necessário utilizar as interrupções padrões de pinos, que ativa em todas mudanças de estado em cada pino habilitado, podemos utilizar os registradores PCINT, descritos abaixo.

O registrador PCICR é responsável por habilitar globalmente as interrupções em cada portal incorporado dentro do ATmega 328.

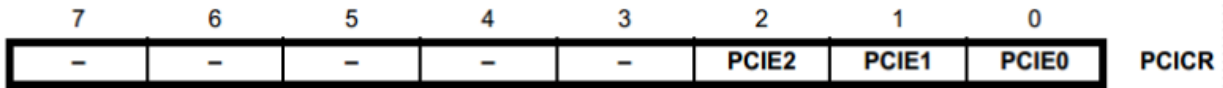
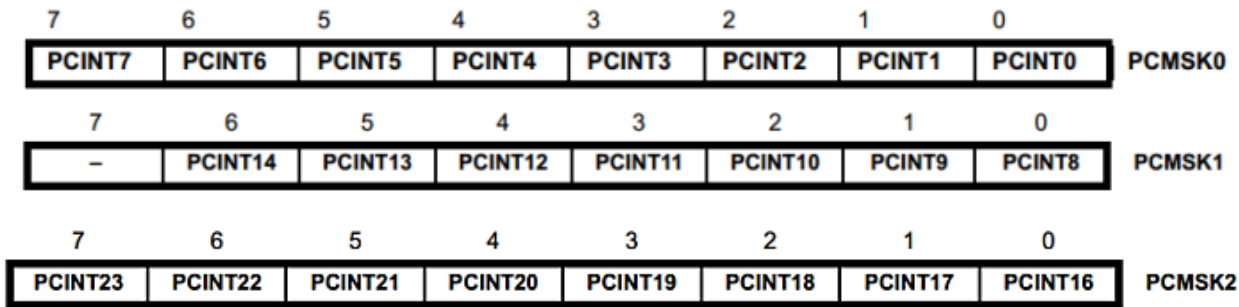


Figura 3 – Registro PCICR

O bit **PCIE0** é responsável por habilitar as interrupções no portal B, o bit **PCIE1** é responsável por habilitar as interrupções no portal C e o bit **PCIE2** é responsável por habilitar as interrupções no portal D. Porém, esse registrador não habilita as interrupções em cada pino do microcontrolador, para tal, é necessário utilizar o registrador PCMSKn, que indica qual pino dentro desses portais que utilizará as interrupções.

Nesse registrador, PCMSK0 habilita as interrupções no portal B, PCMSK1 habilita as interrupções no portal C e PCMSK2 habilita as interrupções no portal D.



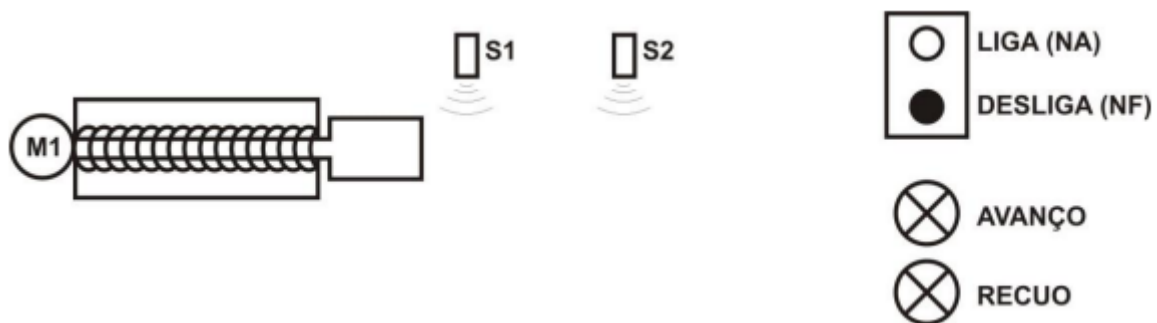
Para fazer uso de qualquer interrupção no ATmega328p é necessário **habilitar o bit de interrupção global**. A linha de código utilizada para isso é: **sei();**. Essa linha acessa o registro que armazena o valor desse bit de configuração e liga o bit.

Para indicar qual função (ou rotina) deve ser executada quando a interrupção ocorrer, usa-se a função **ISR** com o vetor da interrupção desejada:

```
ISR(VETOR)
{
...
    // Rotina de interrupção
...
}
```

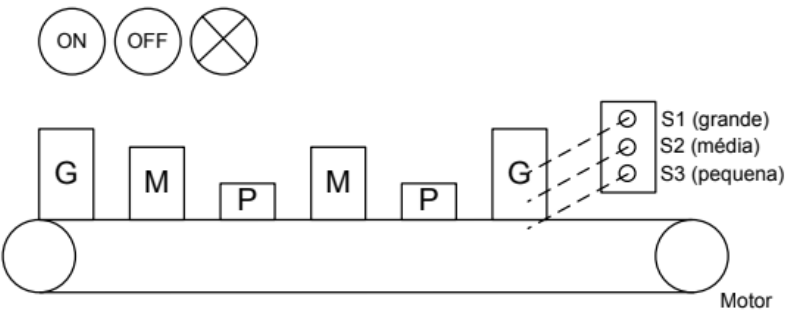
Parte prática:

1. (Fácil) Elaborar um firmware para acionar um buzzer (PB4) sempre que uma mudança de estado aconteça nos seguintes pinos:
 - a. PB2
 - b. PC4
 - c. PD6
 - d. PD2
2. (Médio) O sistema apresentado na figura abaixo foi desenvolvido para identificar o avanço e o recuo de um pistão acionado por um sistema pneumático M1 (ON/OFF). Elabora um firmware seguindo o descritivo apresentado abaixo:
 - a. Ao pressionar o botão LIGA/NA o sistema pneumático é ligado e o pistão avança.
 - b. Ao pressionar o botão DESLIGA/NF o sistema pneumático é desligado e o pistão recua pelo efeito de uma mola;
 - c. Quando o sensor 1 (Interrupção de mudança de estado) for acionado ($S1 = 1$), a indicação de avanço deverá ser ligada ($AVANÇO = 1$) e deverá ser desligada assim que o pistão acionar o sensor 2 ($S2 = 1$) ou desacionar o sensor 1 ($S1 = 0$);
 - d. Quando o sensor 2 (Interrupção de mudança de estado) for acionado ($S2 = 1$) o sistema pneumático deverá ser desligado e uma indicação de recuo deverá ser ligada ($RECUO = 1$) e deverá ser desligada ($RECUO = 0$) assim que o sensor 1 for desacionado ($S1 = 0$).



Entradas / Saídas	GPIO Utilizado
Botão LIGA	PC0
Botão DESLIGA	PC1
S1	PB0
S2	PB1
AVANÇO	PD1
RECUO	PD2
M1	PD3

3. (Difícil) Elaborar um firmware para controlar uma esteira movimentada pelo motor M1 que transporta caixas de três tamanhos (grande, média e pequena) que sensibilizam três sensores ópticos S1, S2 e S3. O processo tem início quando o botão LIGA (NA) é acionado, e **interrompido** pelo botão DESLIGA (NF). A seleção do tipo de caixa, é feita pela chave CH1, como mostra a figura abaixo. Assim, por exemplo, caso seja selecionado caixas pequenas ou médias, o sistema funciona normalmente. Porém quando se passa uma caixa grande, identificada pelo sensor S1, todo o sistema deve parar e um alarme deve piscar com frequência de 5 Hz, além de ativar um buzzer. O alarme é desligado somente pressionando o botão DESLIGA



Entradas / Saídas	GPIO Utilizado
Botão LIGA	PB0
Botão DESLIGA	PD3
S1	PB1
S2	PB2
S3	PD2
M1	PD1
ALARME	PD4
BUZZER	PD5