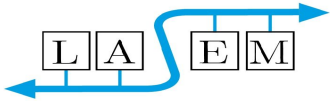
Universidade Federal do Rio Grande do Norte
Instituto Metrópole Digital - IMD
Núcleo de Pesquisa e Inovação em Tecnologia da Informação - NpitI

Programação de Microcontroladores PIC - Dia 02

Ministrante: Fellipe Augusto



Roteiro

- Interrupções
  - Conceitos
  - Exemplo (assembly)
- Timers
  - Conceito
  - HandsOn (assembly e C)
- Conversor Analógico Digital
  - Conceitos
  - Hands-on (C)

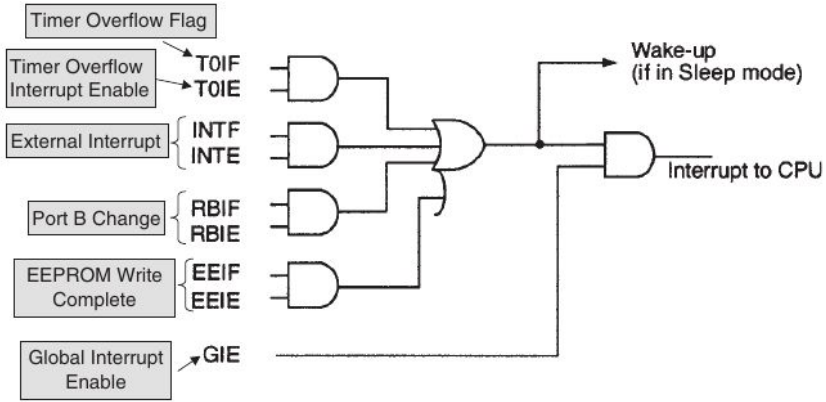
Blink Tradicional

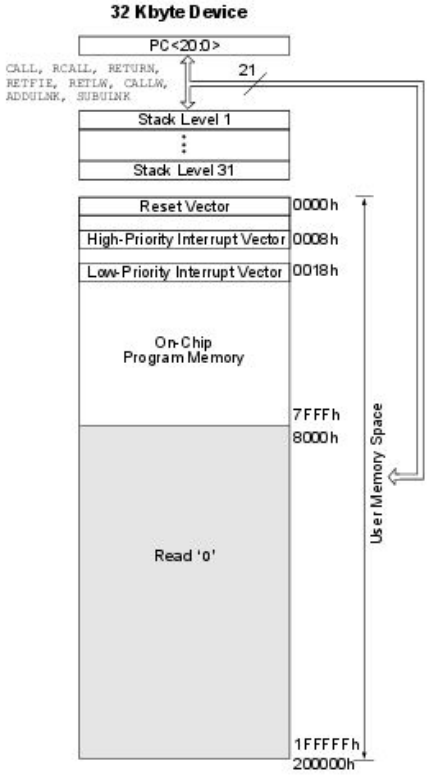
- Como fazer multitasking dessa forma?

```
11  void main(void) {
12      TRISB0=0;
13      while(1){
14      RB0=0;
15      __delay_ms(200);
16      RB0=1;
17      __delay_ms(200);
18      }
19      return;
20  }
```

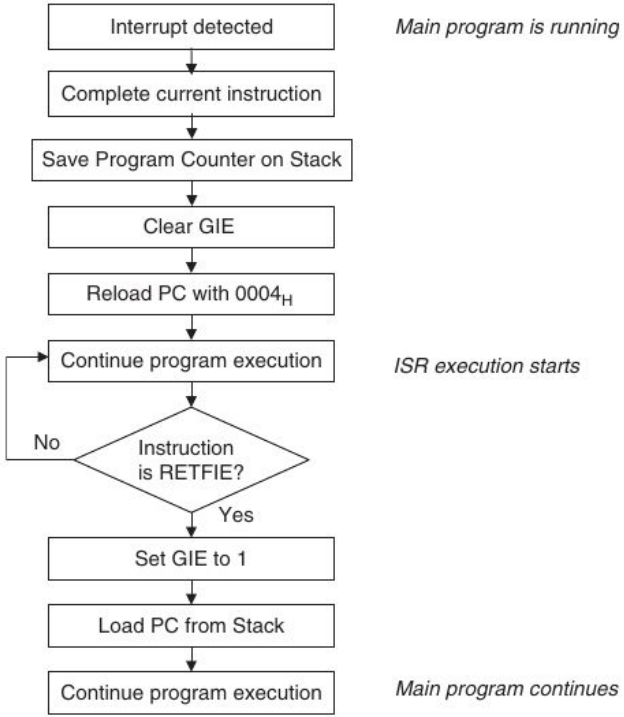
Interrupções: O distúrbio da ordem

- Um Alerta a CPU de que um evento significativamente importante ocorreu





Fluxograma de uma interrupção

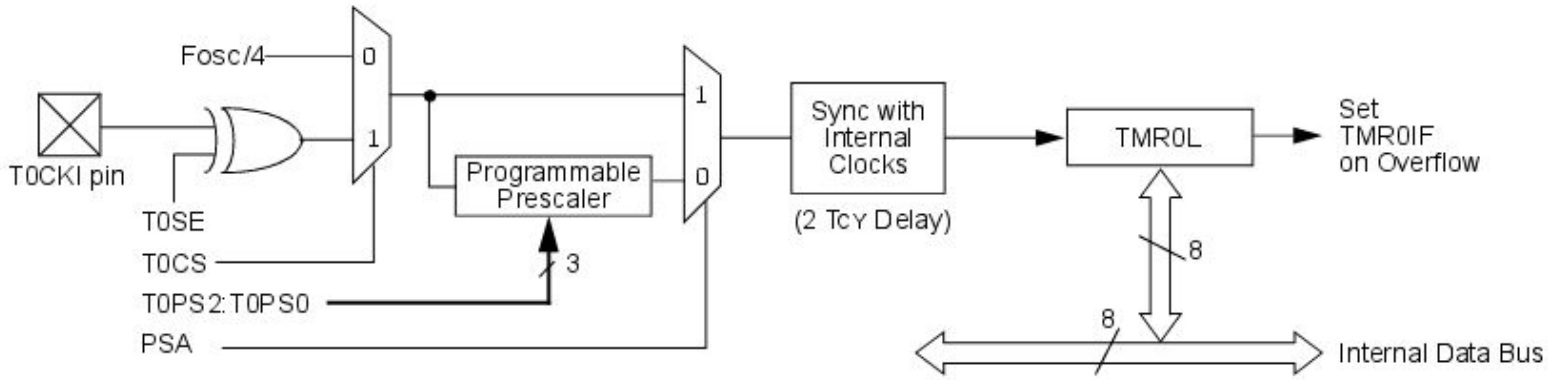


Exemplo de aplicação de Interrupção

- Press Button - Acionar um LED quando o botão for pressionado (detecção do pressionamento via interrupção)
- Configurar interrupções:
  - Registradores INTCON (pag 103), INTCON2 (pag 104), INTCON3 (pag 105)

Timers

- Nosso modelo de PIC (PIC18F4550) possui quatro Timers: Timer 0, Timer 1, Timer2 e Timer3
- Contador e temporizador
  - Temporização do Sistema
  - Utilização no módulo de captura e comparação de sinais (Timer1 e Timer3)
  - Utilização no módulo PWM (Timer2)



Note: Upon Reset, Timer0 is enabled in 8-bit mode with clock input from T0CKI maximum prescale.

Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Reset Values on page
TMR0L	Timer0 Register Low Byte								54
TMR0H	Timer0 Register High Byte								54
INTCON	GIE/GIEH	PEIE/GIEL	TMR0IE	INT0IE	RBIE	TMR0IF	INT0IF	RBIF	53
INTCON2	$\overline{\text{RBPU}}$	INTEDG0	INTEDG1	INTEDG2	—	TMR0IP	—	RBIP	53
T0CON	TMR0ON	T08BIT	T0CS	T0SE	PSA	T0PS2	T0PS1	T0PS0	54
TRISA	—	TRISA6(1)	TRISA5	TRISA4	TRISA3	TRISA2	TRISA1	TRISA0	56

Registradores Associados ao Timer 0

Configuração específica do Timer0: T0CON (pag 109)

Hands-On: Timer0

- Fazer o download do header file contendo o protótipo das funções de configuração (timer0lib.h)
- Implementar o source file (timer0lib.c)
- Implementar o arquivo main.c:
  - Desenvolver um BLINK (simular no Proteus) utilizando interrupção do Timer 0
  - Frequência: 1 Hz
  - Testar no kit
  - Dica: usar o material adicional de configuração de Timers

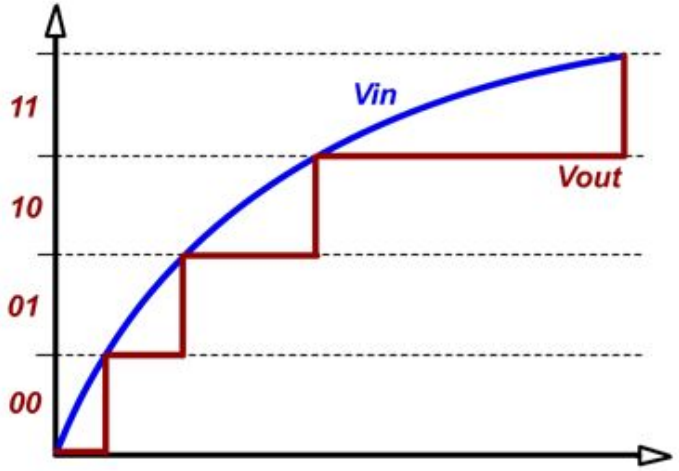
Desafio

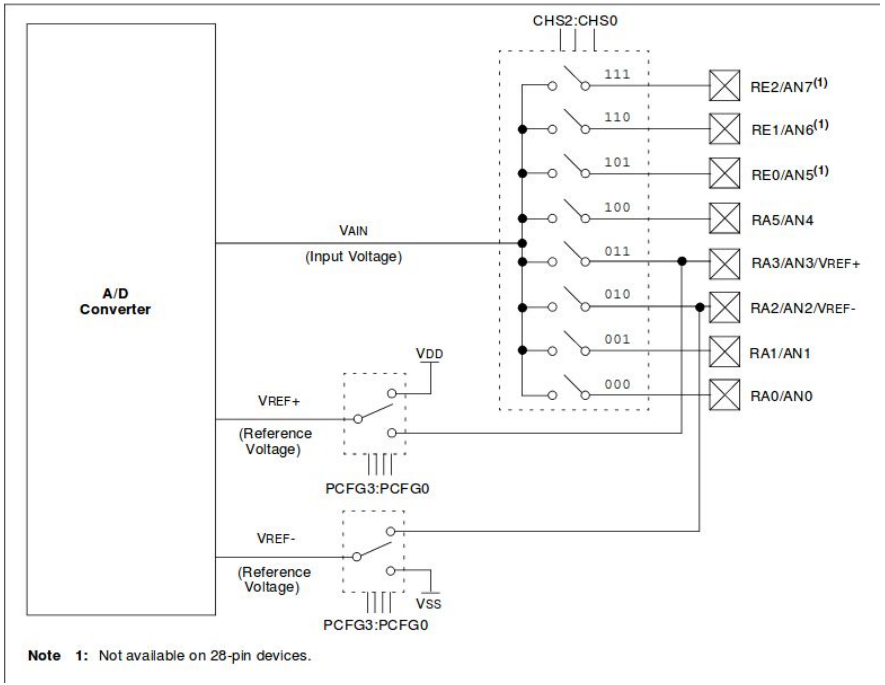
A partir da biblioteca desenvolvida, implementar um projeto de blink com dois leds em frequências diferentes

- Utilize a interrupção do Timer0

Conversor Analógico Digital (ADC)

- Conversão de sinais analógicos para binários de 10 bits





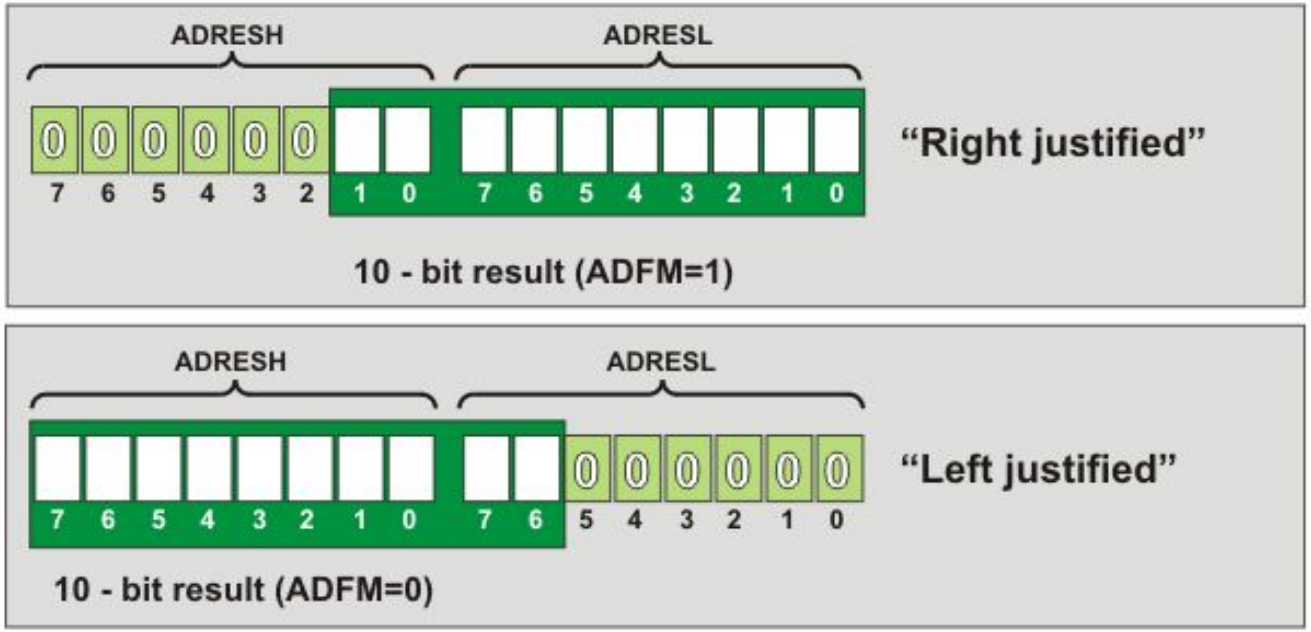
Que taxa de conversão escolher?

AD Clock Source (T_{AD})		Assumes T_{AD} Min. = 0.8 µs
Operation	**ADCS2:ADCS0**	**Maximum Fosc**
2 Tosc	000	2.50 MHz
4 Tosc	100	5.00 MHz
8 Tosc	001	10.00 MHz
16 Tosc	101	20.00 MHz
32 Tosc	010	40.00 MHz
64 Tosc	110	48.00 MHz
RC(2)	x11	1.00 MHz(1)

Registradores de Configuração

- ADCON0
  - O que ele faz?
- ADCON1
  - O que ele faz?
- ADCON2
  - O que ele faz?

Formatação do Valor Digital



HandsOn - ADC

- Fazer o download do header file adclib.h
- Implementar o source file adclib.c
- Implementar a função main.c
  - Ler um valor analógico do ADC e refletir tal valor nos LEDs
  - Uso do kit de desenvolvimento

Desafio

A partir da biblioteca desenvolvida, implementar um projeto em que, após determinado valor limiar lido pelo ADC, habilitar um blink de alerta

- Utilize a interrupção do Timer0;
- Utilizar a biblioteca para display de sete segmentos;
- Utilizar a biblioteca de manipulação dos GPIOS