**MICROCONTROLADORES**

**Exercício 2**

**Nome 1**: Victor Elioenay Santos Narciso

**Nome 2**: Luiz Gustavo Chinelato Setten

**Registradores Especiais**:

# INTCON-

Configuração: 1111000.

Configurado para habilitar a interrupção RB0/INT e a interrupção do TMR0.

# TRISA-

Configuração: 00000000.

Configurado para habilitar todas as os bits de PORTA como saídas.

# TRISB-

Configuração: 00000101.

Configurado para habilitar os bits RB0 e RB2 como entrada. RB0 utilizado para ativar a interrupção, e RB2 utilizado para o contador.

# OPTION\_REG-

Configuração: 11000000.

Configurado para desabilitar os pull-up’s do PORTB, habilita a interrupção do RB0/INT por borda de subida e configura o prescaler do TMR0 em 1:8.

# Código adicionado para o tratamento da interrupção

**ORG** 0x04 ;Endereço inicial da interrupção

**MOVWF W\_TEMP** ;Move o estado atual do acumulador para o W\_TEMP

**SWAPF STATUS, W** ;Move o estado atual do STATUS para o acumulador, com a sequência de BITS invertida

;Necessita de usar o SWAPF, pois o MOVFW altera o estado do STATUS

**MOVWF STATUS\_TEMP** ;Move do acumulador para o STATUS\_TEMP

**BTFSS INTCON,INTF** ;Verifica o nível lógico do RB0. Caso 1, trata a interrupção do RB0, caso 0, trata a interrupção do TMR0

**GOTO TIMER** ;Jump para tratamento da interrupção do TMR0

**GOTO LED** ;Jump para tratamento da interrupção do RB0

## FIM

**SWAPF STATUS\_TEMP, W** ;Move o estado de STATUS\_TEMP para o acumulador, desinvertendo o seu estado

**MOVWF STATUS** ;Move do acumulador para o STATUS

**SWAPF W\_TEMP, F** ;Inverte o estado de W\_TEMP e armazena no própio

**SWAPF W\_TEMP, W** ;Move o estado de W\_TEMP para o acumulador

;Aparti daqui tanto o STATUS como o acumulador retornaram ao estado que estavem antes de entrarem na interrupção

**BCF INTCON, 1** ;Limpa o bit no INTCON que sinaliza que ocorreu a interrupção no RBO/INT

**RETFIE** ;Retorna da interrupção

**LED**

**BTFSS PORTB,7** ;Verifica o nível lógico do RB7. Caso 1, apaga, caso 0, acende

**GOTO ACENDE** ;Jump para onde será setado nível lógico alto no RB7

**GOTO APAGA** ;Jump para onde será setado nível lógico baixo no RB7

## APAGA

**BCF PORTB,7** ;Seta RB7 com nível lógico baixo

**GOTO FIM** ;Pula para FIM

## ACENDE

**BSF PORTB,7** ;Seta RB7 com nível lógico alto

**GOTO FIM** ;Pula para FIM

**TIMER**

**BTFSS PORTB,6** ;Verifica o nível lógico do RB6. Caso 1, vai para alto, caso 0, vai para baixo

**GOTO ALTO**  ;Pula para ALTO

**GOTO BAIXO** ;Pula para BAIXO

**ALTO**

**BSF PORTB,6** ;Seta o nível de RB6 para alto

**BCF INTCON,T0IF** ;Limpa a flag da interrupção de TMR0

**MOVLW .256-.123** ;Move o valor inicial do contador de TMR0 para o acumulador

**MOVWF TMR0** ;Move o valor do acumulador para TMR0

**GOTO FIM** ;Vai para FIM

**BAIXO**

**BCF PORTB,6** ;Seta o nível de RB6 para baixo

**BCF INTCON,T0IF** ;Limpa a flag da interrupção de TMR0

**MOVLW .256-.123** ;Move o valor inicial do contador de TMR0 para o acumulador

**MOVWF TMR0** ;Move o valor do acumulador para TMR0

**GOTO FIM** ;Vai para FIM

# Diagrama do Circuito Simulado no Proteus

