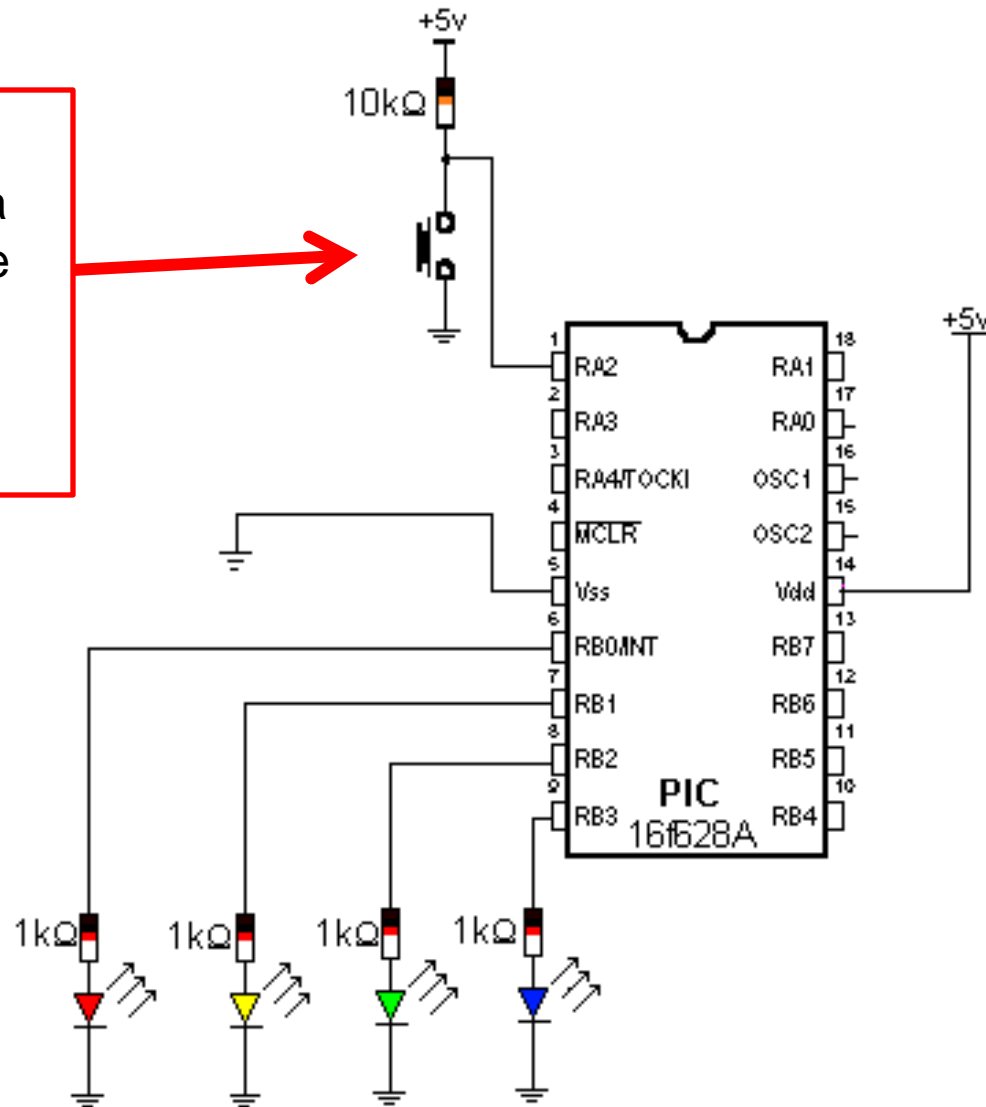


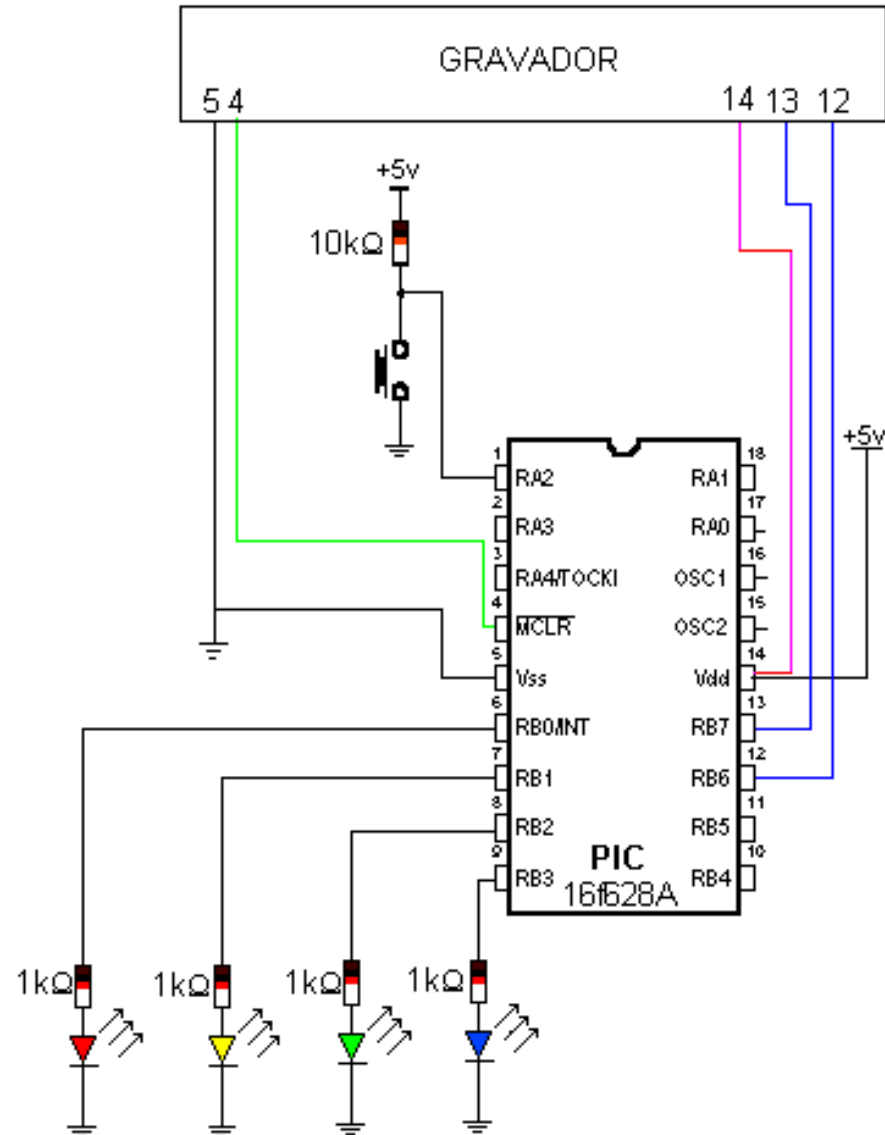
Circuito a ser utilizado para a aula 2

ATENÇÃO

A chave está normalmente ligada ao 5v (ou 1) e não ao 0.



Circuito a ser utilizado para a aula 2 (com o gravador)



Programa a ser utilizado para a aula 2

```
#INCLUDE <P16F628A.INC>                ;ARQUIVO PADRÃO MICROCHIP PARA 16F628A
__CONFIG H'3F10'

#DEFINE BANK0    BCF STATUS,RP0        ;SETA BANK 0 DE MEMÓRIA
#DEFINE BANK1    BSF STATUS,RP0        ;SETA BANK 1 DE MAMÓRIA
CBLOCK 0x20      ;ENDEREÇO INICIAL DA MEMÓRIA DE USUÁRIO
ENDC           ;FIM DO BLOCO DE MEMÓRIA
#DEFINE BOTAO    PORTA,2                ;PORTA DO BOTÃO      ; 0 -> PRESSIONADO
                                           ; 1 -> LIBERADO
#DEFINE LED      PORTB,0                ;PORTA DO LED        ; 0 -> APAGADO
                                           ; 1 -> ACESO
ORG 0x00        ;ENDEREÇO INICIAL DE PROCESSAMENTO
GOTO INICIO

CLRF PORTA      ;LIMPA O PORTA
CLRF PORTB      ;LIMPA O PORTB
BANK1           ;ALTERA PARA O BANCO 1
MOVLW B'00000100'
MOVWF TRISA     ;DEFINE RA1 COMO ENTRADA E DEMAIS COMO SAÍDAS
MOVLW B'00000000'
MOVWF TRISB     ;DEFINE TODO O PORTB COMO SAÍDA
MOVLW B'00000000'
MOVWF INTCON    ;TODAS AS INTERRUPÇÕES DESLIGADAS
BANK0           ;RETORNA PARA O BANCO 0
MOVLW B'00000111'
MOVWF CMCON     ;DEFINE O MODO DO COMPARADOR ANALÓGICO
                ; 007 PARA ENTRADAS DIGITAIS
```

MAIN

BTFSC	BOTAO	;O BOTÃO ESTÁ PRESSIONADO?
GOTO	BOTAO_LIB	;NÃO, ENTÃO TRATA BOTÃO LIBERADO
GOTO	BOTAO_PRES	;SIM, ENTÃO TRATA BOTÃO PRESSIONADO

BOTAO_LIB

BCF	LED	;APAGA O LED
GOTO	MAIN	;RETORNA AO LOOP PRINCIPAL

BOTAO_PRES

BSF	LED	;ACENDE O LED
GOTO	MAIN	;RETORNA AO LOOP PRINCIPAL

END	;OBRIGATÓRIO
-----	--------------

Entendendo o programa

```
#DEFINE  BANK0    BCF STATUS,RP0           ;SETA BANK 0 DE MEMÓRIA

#DEFINE  BANK1    BSF STATUS,RP0           ;SETA BANK 1 DE MAMÓRIA

#DEFINE  BOTAO    PORTA,2                  ;PORTA DO BOTÃO
                                           ; 0 -> PRESSIONADO
                                           ; 1 -> LIBERADO

#DEFINE  LED      PORTB,0                  ;PORTA DO LED
                                           ; 0 -> APAGADO
                                           ; 1 -> ACESO
```

A diretiva **#DEFINE** apenas irá substituir no programa principal, os termos utilizados pelos padrões do PIC.

Exemplo:

- se no programa digitarmos **LED**, ele será substituído por **PORTB, 0**
- se digitarmos **BOTAO**, ele será substituído por **PORTA, 2**

Entendendo o programa

```
ORG      0x00          ;ENDEREÇO INICIAL DE PROCESSAMENTO
GOTO     INICIO
```

A instrução **ORG** apenas indica o início do programa.
O **GOTO** leva a execução para a linha **INICIO**.

Isso se faz necessário porque quando uma interrupção acontece o programa é desviado para o endereço 0x04.
Assim, nosso programa permite que, posteriormente, possamos inserir algum código para controle de alguma interrupção no endereço 0x04.

Entendendo o programa

INICIO

```
CLRF    PORTA    ;LIMPA O PORTA
CLRF    PORTB    ;LIMPA O PORTB

BANK1    ;ALTERA PARA O BANCO 1

MOVLW   B'00000100'
MOVWF   TRISA    ;DEFINE RA2 COMO ENTRADA E DEMAIS COMO SAÍDAS

MOVLW   B'00000000'
MOVWF   TRISB    ;DEFINE TODO O PORTB COMO SAÍDA

MOVLW   B'00000000'
MOVWF   INTCON   ;TODAS AS INTERRUPÇÕES DESLIGADAS

BANK0    ;RETORNA PARA O BANCO 0

MOVLW   B'00000111'
MOVWF   CMCON    ;DEFINE O MODO DO COMPARADOR ANALÓGICO
           ; 007 PARA ENTRADAS DIGITAIS
```

Essas são instruções para configuração dos pinos. Quais serão utilizados como entradas e quais como saídas.

TRISA(B) está no banco1 e configura cada pino dos portos A(B) como entrada(1) ou saída(0).

INTCON se refere às interrupções, todo em zero significa interrupções desligadas.

CMCON configura os pinos de entrada, se em 7(111) as entradas funcionam como entradas digitais (sensíveis a 5 volts ou 0 volts).

Entendendo o programa

MAIN

```
BTFSC    BOTAO           ;O BOTÃO ESTÁ PRESSIONADO?  
GOTO     BOTAO_LIB       ;NÃO, ENTÃO TRATA BOTÃO LIBERADO  
GOTO     BOTAO_PRES      ;SIM, ENTÃO TRATA BOTÃO PRESSIONADO
```

BOTAO_LIB

```
BCF      LED             ;APAGA O LED  
GOTO     MAIN            ;RETORNA AO LOOP PRINCIPAL
```

BOTAO_PRES

```
BSF      LED             ;ACENDE O LED  
GOTO     MAIN            ;RETORNA AO LOOP PRINCIPAL
```

```
END                      ;OBRIGATÓRIO
```

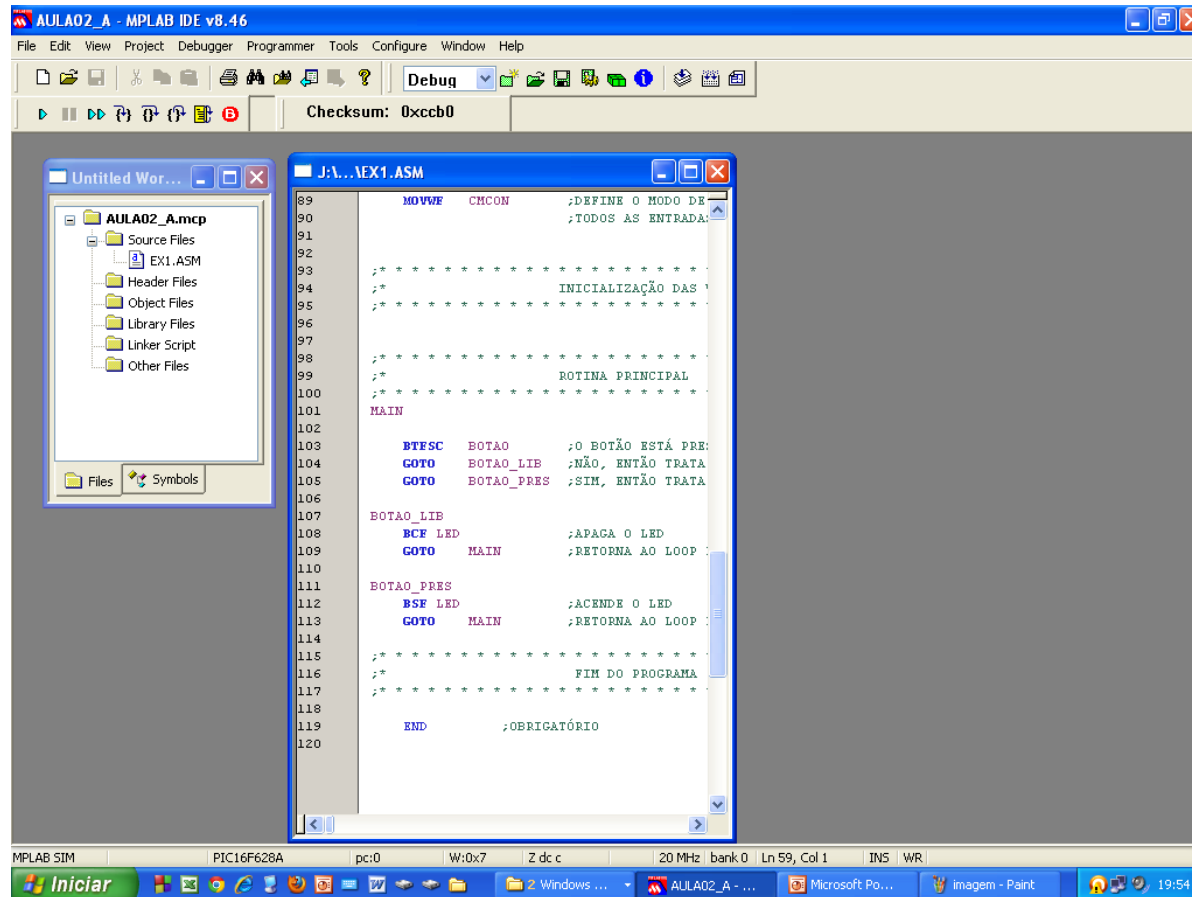
Observar que a instrução **BTFSC** significa para testar um determinado bit do argumento e pular a próxima linha se ele for zero.

A instrução **BCF** “reseta” (leva a 0) um determinado bit indicado pelo argumento.

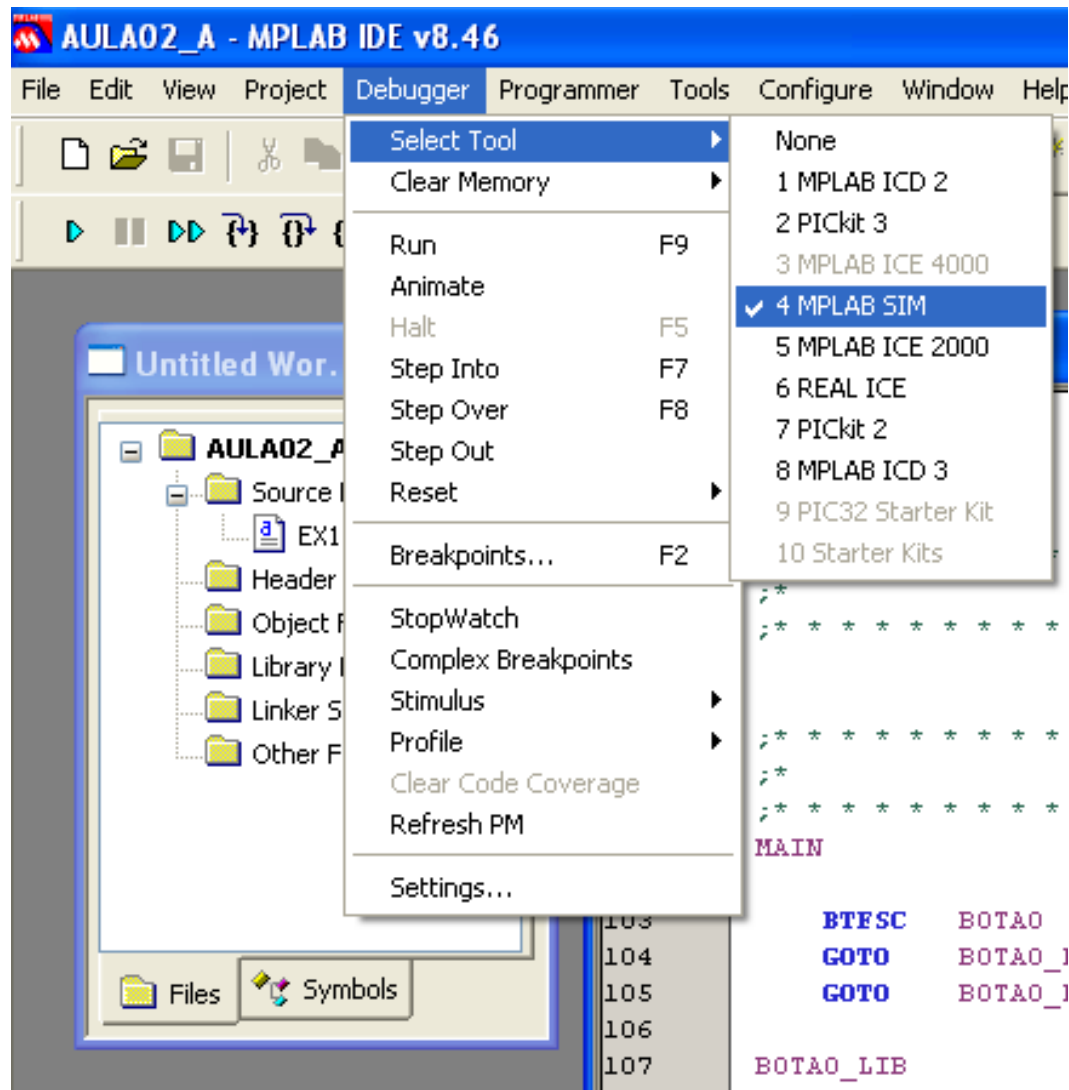
A instrução **BSF** “seta” (leva a 1) um determinado bit indicado pelo argumento.

Como usar o editor de estímulo e a simulação visual no MPLAB

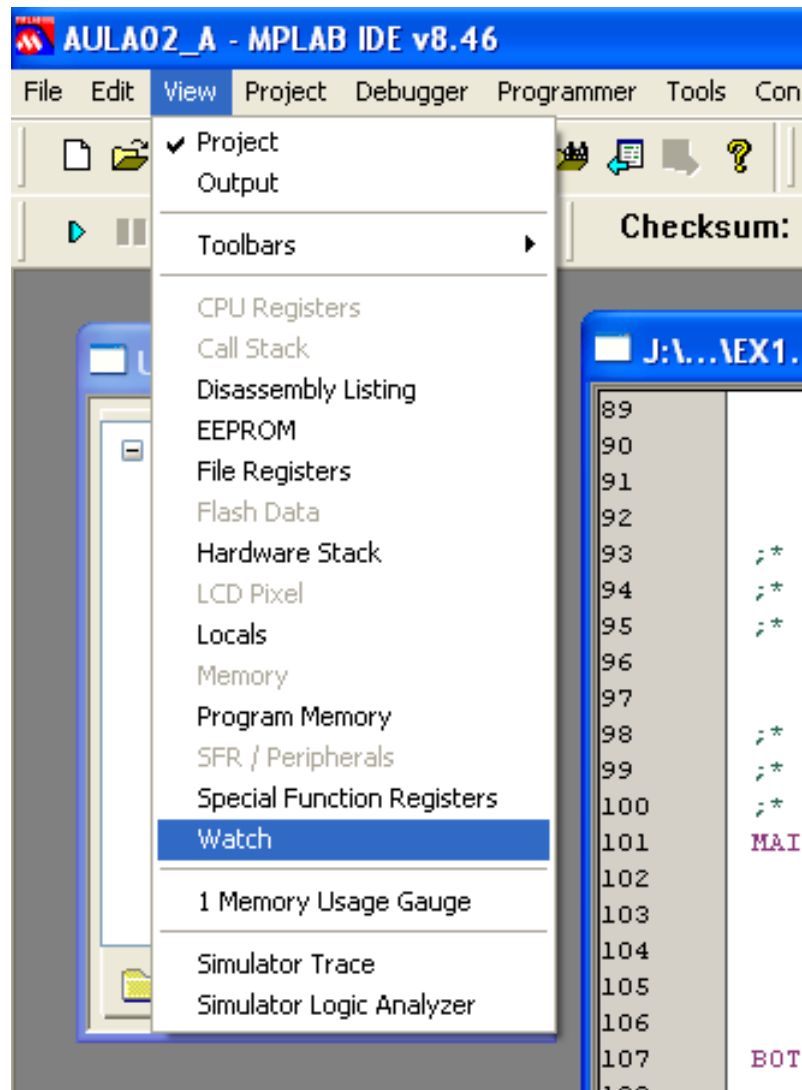
1) Crie um projeto e inclua o arquivo anterior (botão e led)



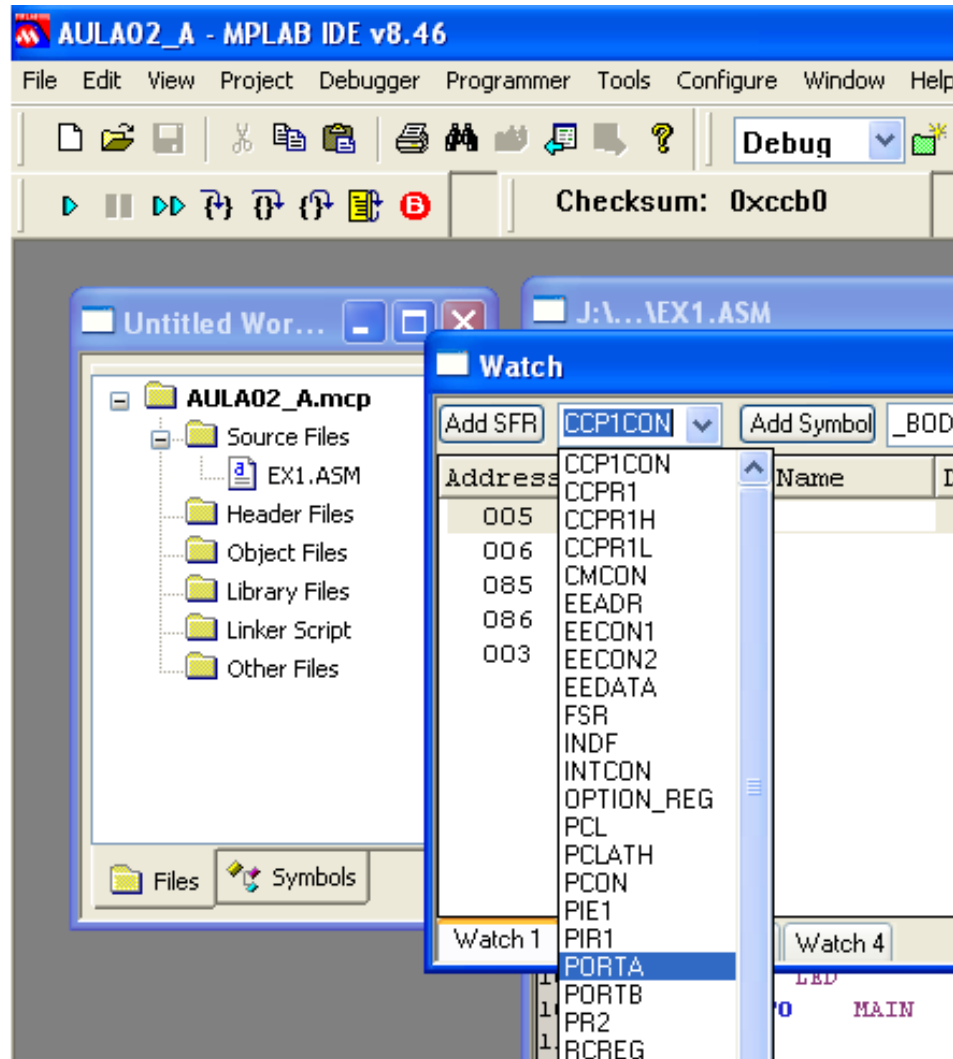
2) Selecione a ferramenta de simulação (MPLAB SIM)



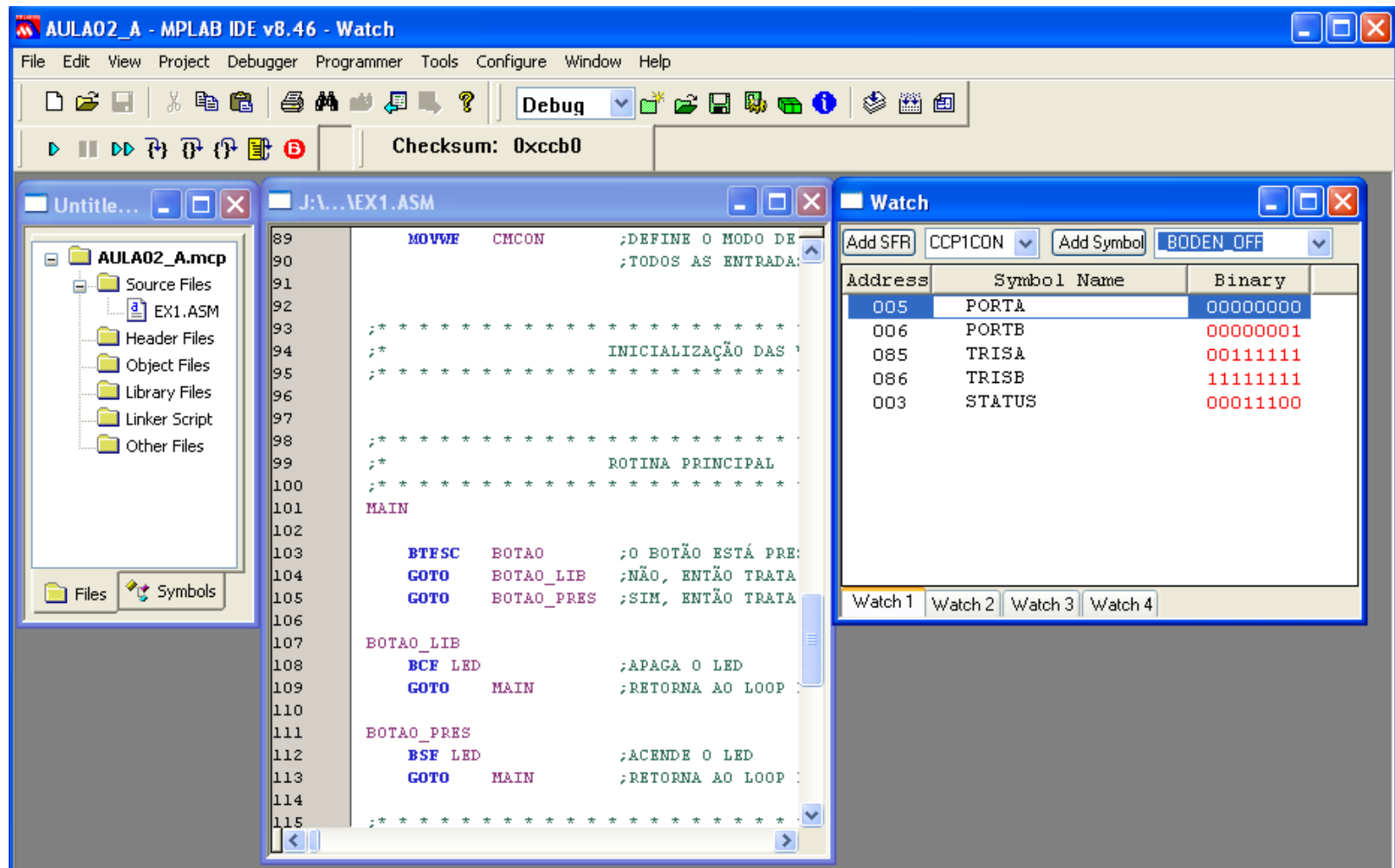
3) Selezione a ferramenta WATCH



- 3) Selecione os registradores a serem observados.
- 4) Escolha os registradores e clique no botão ADD SFR.

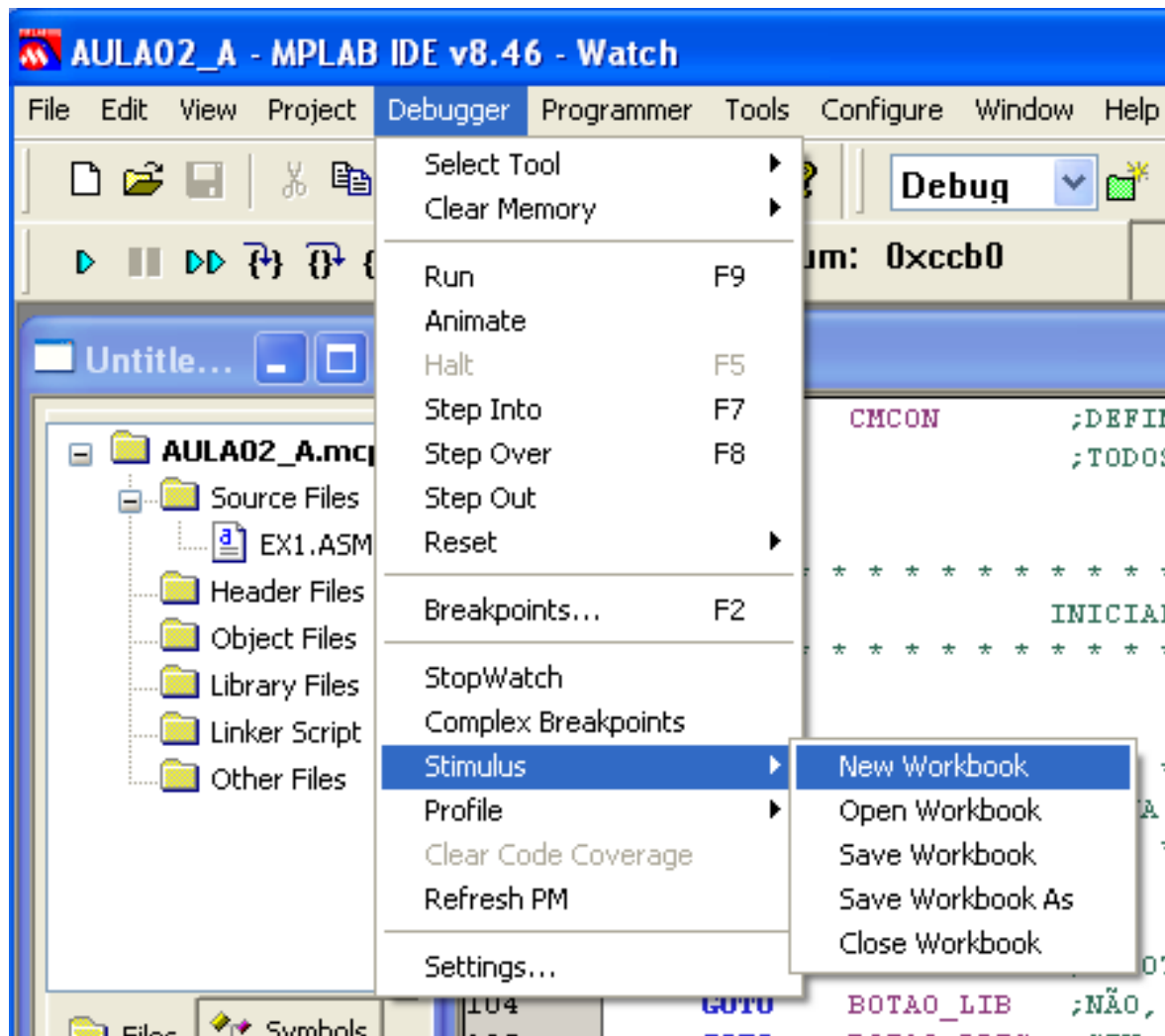


3) Selecione os formatos (hexa, dec., bin.)



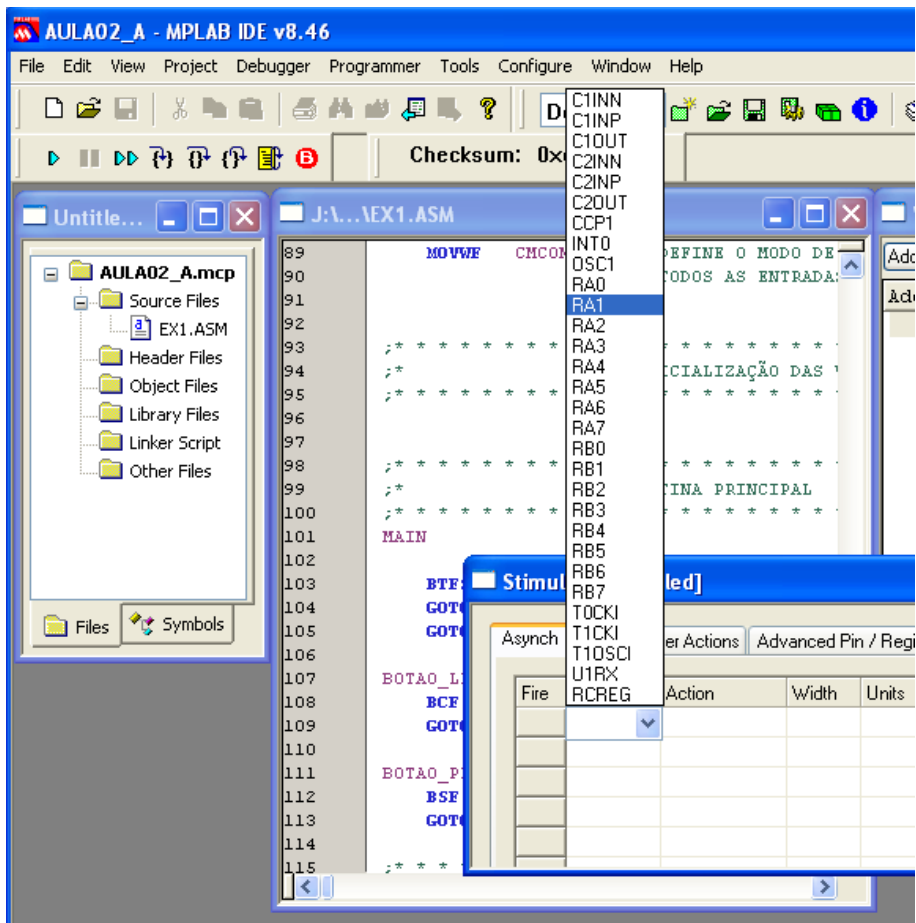
Exemplo de como poderá ficar a sua tela

3) Seleccione o editor de estímulo

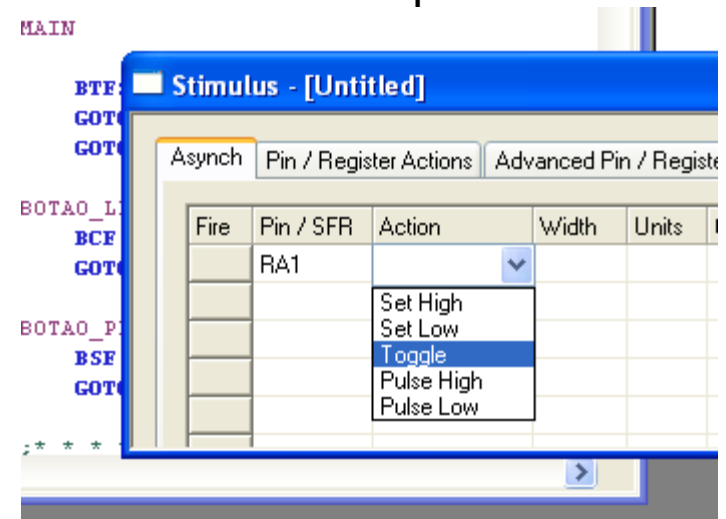


- 3) De acordo com o programa, escolha qual será o registrador afetado, no nosso caso, estaremos alterando o botão em RA1.
- 4) Escolha a forma de alteração da entrada, no caso, modo toggle.

3



4



Tela de simulação até o momento

AULA02_A - MPLAB IDE v8.46 - Stimulus - [Untitled]

File Edit View Project Debugger Programmer Tools Configure Window Help

Checksum: 0xccb0 Debug

Untitled... J:\...\EX1.ASM Watch

Files **Symbols**

- AULA02_A.mcp
 - Source Files
 - EX1.ASM
 - Header Files
 - Object Files
 - Library Files
 - Linker Script
 - Other Files

```

89      MOVWF    CMCON      ;DEFINE O MODO DE
90                          ;TODAS AS ENTRADA:
91
92
93      ; * * * * *
94      ; *          INICIALIZAÇÃO DAS
95      ; * * * * *
96
97
98      ; * * * * *
99      ; *          ROTINA PRINCIPAL
100     ; * * * * *
101     MAIN
102
103     BTFSC    BOTAO      ;O BOTÃO ESTÁ PRE:
104     GOTO     BOTAO_LIB   ;NÃO, ENTÃO TRATA
105     GOTO     BOTAO_PRES  ;SIM, ENTÃO TRATA
106
107     BOTAO_LIB
108     BCF LED    ;APAGA O LED
109     GOTO     MAIN        ;RETORNA AO LOOP
110
111     BOTAO_PRES
112     BSF LED    ;ACENDE O LED
113     GOTO     MAIN        ;RETORNA AO LOOP
114
115     ; * * * * *

```

Watch

Add SFR CCP1CON Add Symbol BODEN OFF

Address	Symbol Name	Binary
005	PORTA	00000000
006	PORTB	00000001
085	TRISA	00111111
086	TRISB	11111111
003	STATUS	00011100

Watch 1 Watch 2 Watch 3 Watch 4

Stimulus - [Untitled]

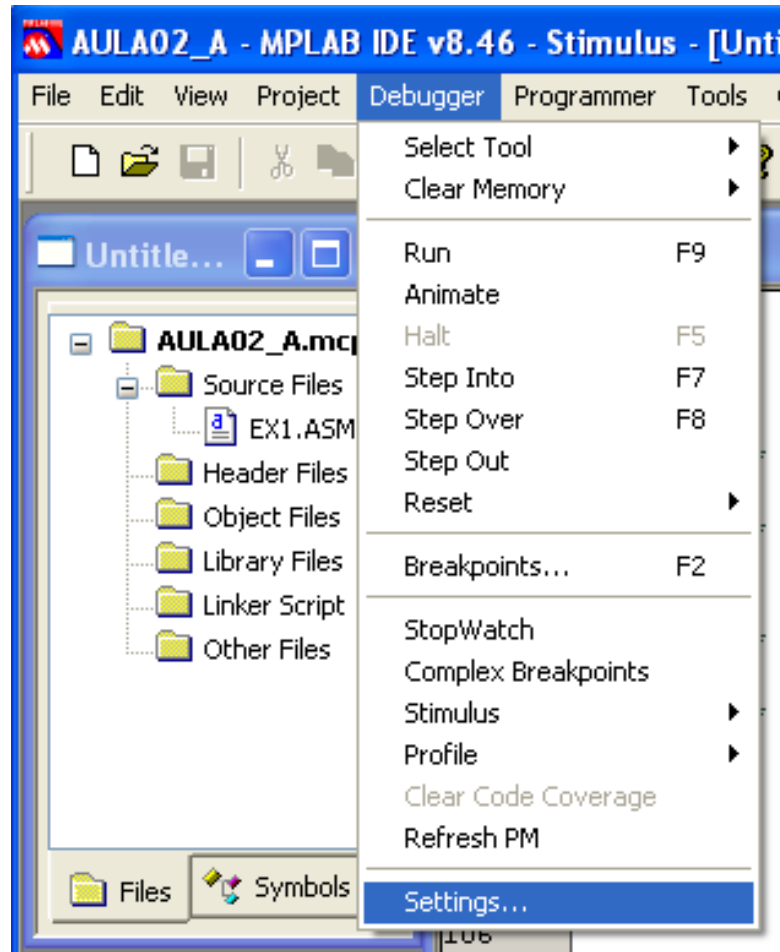
Asynch Pin / Register Actions Advanced Pin / Register Clock S

Fire	Pin / SFR	Action	Width	Units	Comments
>	RA1	Toggle			

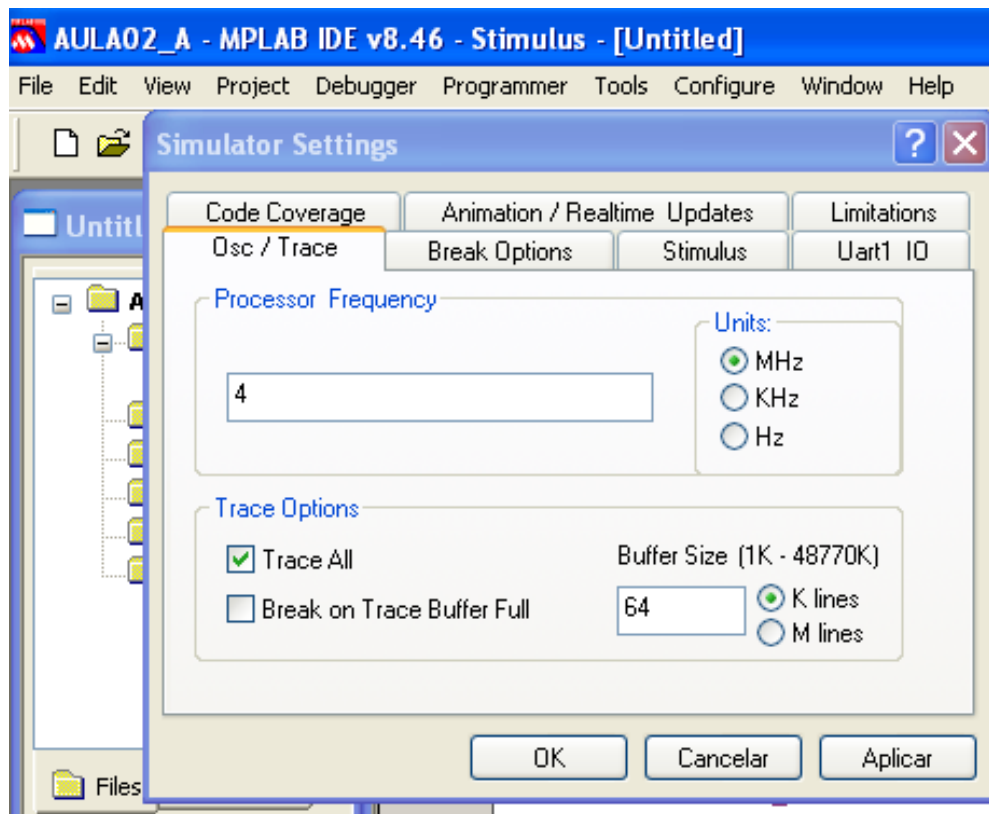
MPLAB SIM PIC16F628A pc:0 W:0x7 Z dc c 20 MHz bank 0 WR

Ajustando a simulação

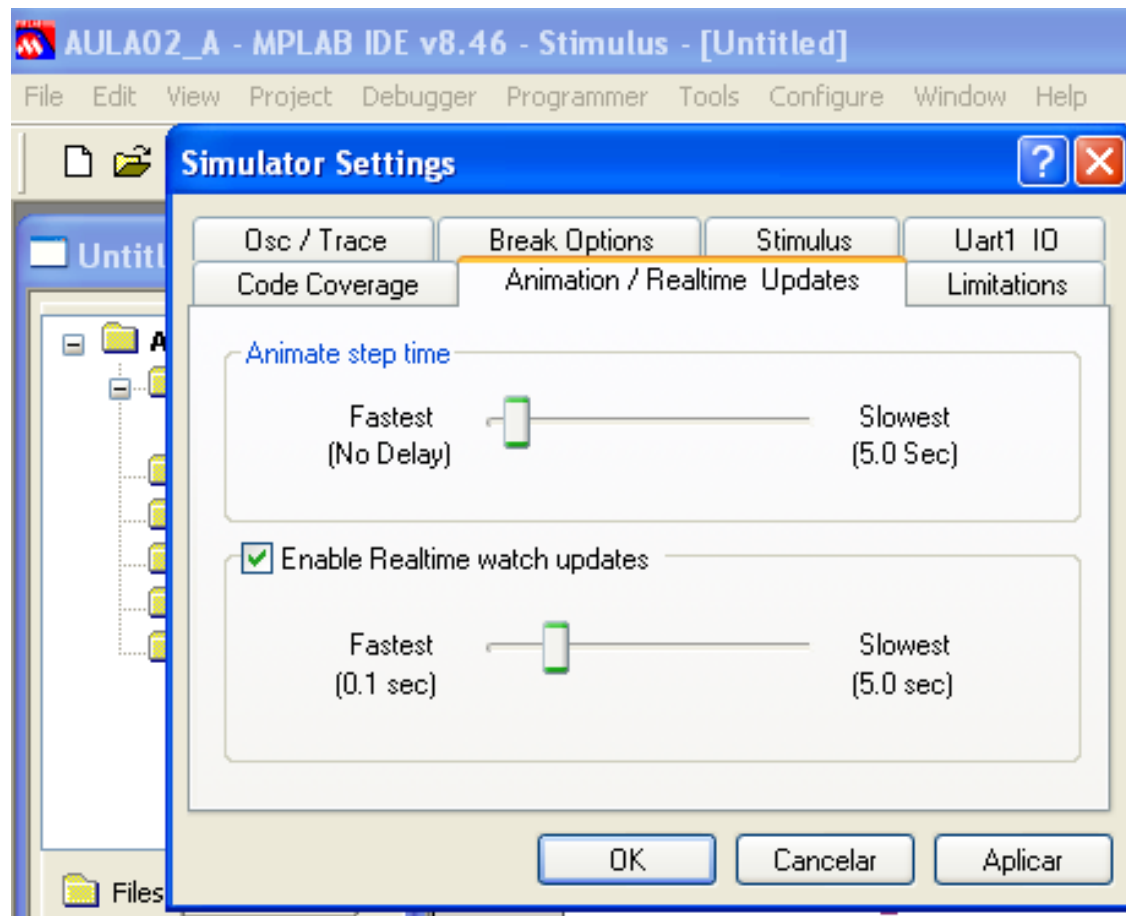
1) Entre no menu settings do Debugger



2) Passe a frequencia para 4 MHz, menu Osc/Trace

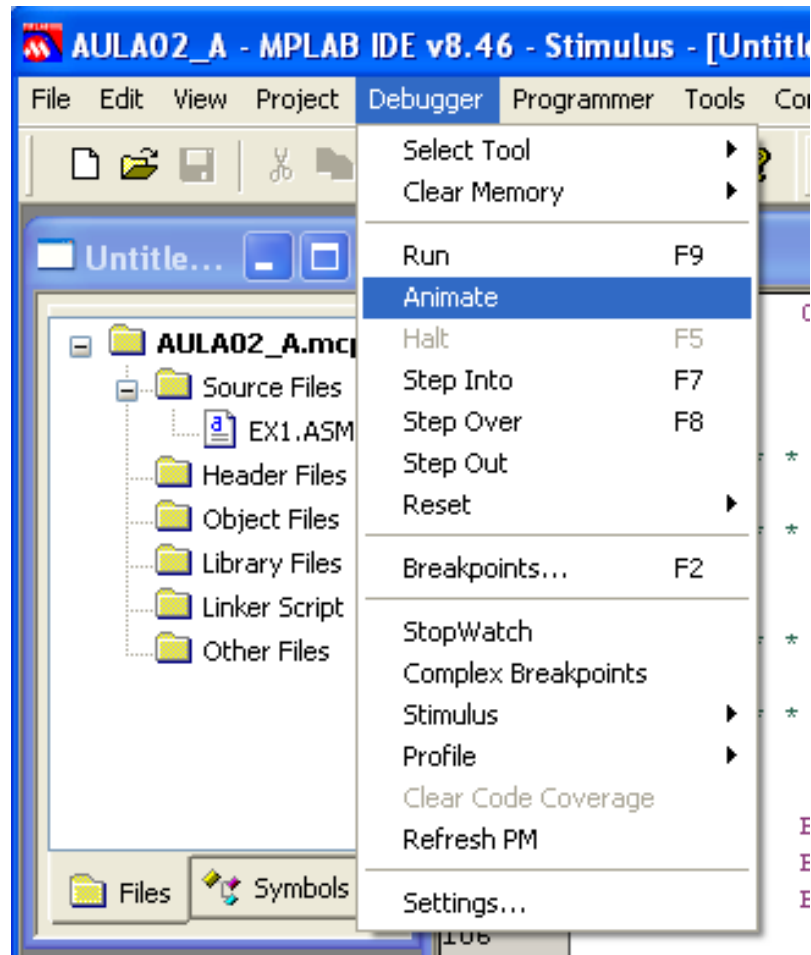


3) Habilite o Enable Realtime with updates, menu Animation/Realtime Updates



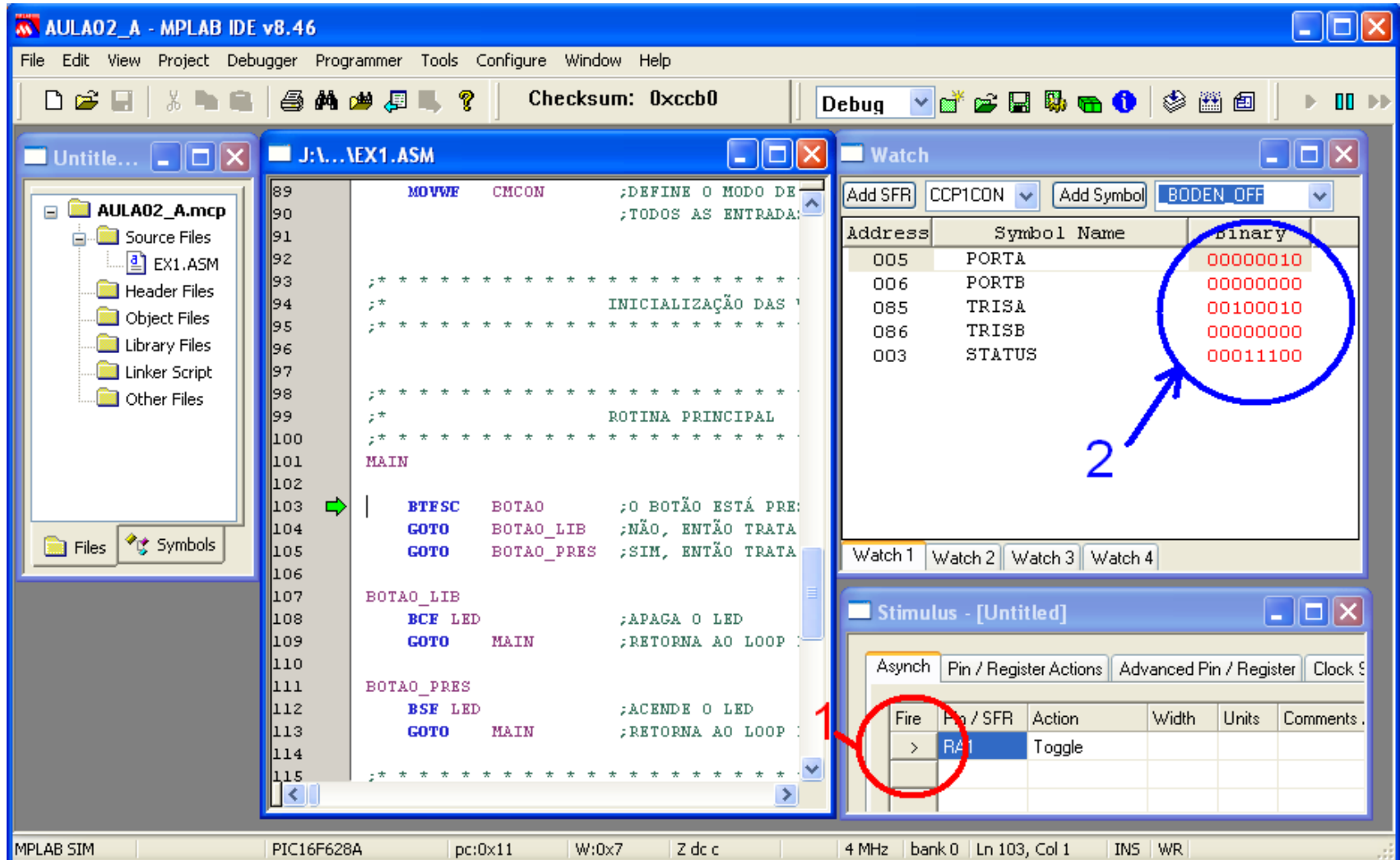
4) Clique em OK

3) Ligue a animação



3) Clique em **1** para ligar e desligar a chave em RA1

4) Observe em **2** a alteração dos bits nos registradores



Analise o seguinte programa

MAIN

BTFSC	BOTAO	;O BOTÃO ESTÁ PRESSIONADO?
GOTO	MAIN	;NÃO, ENTÃO ESPERA PRESSIONAR
GOTO	BOTAO_PRES1	;SIM, ENTÃO TRATA BOTÃO PRESSIONADO

BOTAO_PRES1

BTFSS	BOTAO	;O BOTÃO AINDA ESTÁ PRESSIONADO?
GOTO	BOTAO_PRES1	;SIM, ENTÃO ESPERA DESLIGAR
BSF	LED	;NÃO, ACENDE O LED
GOTO	TESTA1	;VAI PARA O PROXIMO TESTE

TESTA1

BTFSC	BOTAO	;O BOTÃO ESTÁ PRESSIONADO?
GOTO	TESTA1	;NÃO, ENTÃO ESPERA PRESSIONAR
GOTO	BOTAO_PRES2	;SIM, ENTÃO TRATA BOTÃO PRESSIONADO

BOTAO_PRES2

BTFSS	BOTAO	;O BOTÃO AINDA ESTÁ PRESSIONADO?
GOTO	BOTAO_PRES2	;SIM, ENTÃO ESPERA DESLIGAR
BCF	LED	;NÃO, ACENDE O LED
GOTO	MAIN	;RETORNA AO LOOP PRINCIPAL

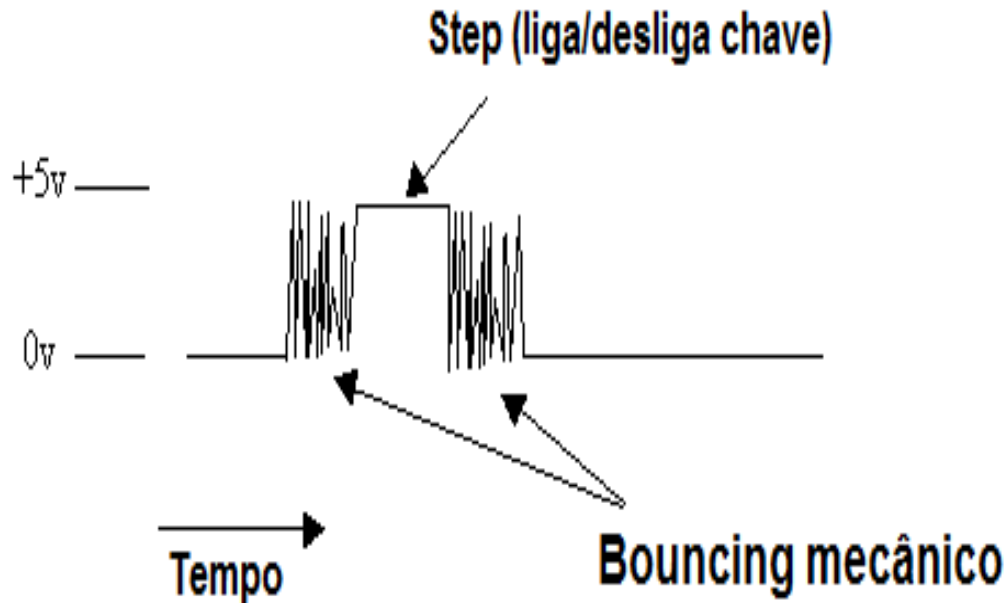
END		;OBRIGATÓRIO
-----	--	--------------

Relatório:

- 1) Crie um programa onde cada vez que pressionarmos a chave, um led ascende e outro apaga. Faça com os dois primeiros leds.
- 2) Repita o exercício anterior usando 3 leds e preservando a sequencia.

Problemas no uso de chaves

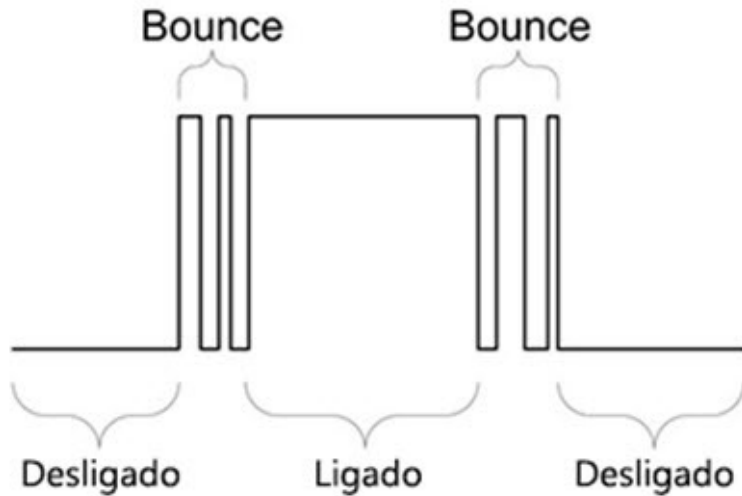
Um dos grandes problemas quando lidamos com chaves chama-se bouncing.



Como uma chave é um dispositivo mecânico, ocorrem oscilações quando uma chave é ligada ou desligada.

Essas oscilações são muito mais lentas que o tempo de avaliação da chave pelo microcontrolador.

Problemas no uso de chaves



Podemos ter o que ocorre na figura ao lado.

Se o Microcontrolador está verificando a chave em um tempo muito pequeno, ele pode entender que a chave está mudando (de desligado para ligado ou ligado para desligado) quando na verdade não está. É apenas a oscilação que ocorre no momento que a chave é ligada ou desligada.

Uma solução em software é imaginar que essa chave irá demorar, por exemplo, 50 ms para estabilizar (esse valor irá depender da qualidade da chave utilizada).