

# Mini-curso MpLab passo-a-passo

MpLab 5.00 (e superiores)

## GUIA PRÁTICO PARA SIMULAÇÃO E ESTUDOS.

VIDAL - Projetos Personalizados  
www.vidal.com.br

### USANDO O MpLab 5.00 E SUPERIOR

#### Introdução e ajustes

O MpLab é um ambiente integrado para o estudo e desenvolvimento com a família PIC de microcontroladores.

Sua principal característica é a total integração de seus módulos com o ambiente Windows, permitindo a fácil cópia de arquivos e trechos de arquivos de um aplicativo para outro.

Para se escrever um programa no MpLab, precisamos antes assimilar o conceito de “projeto”.

#### O “Projeto” no MpLab

Entende-se por projeto um conjunto de arquivos e informações que diz ao MpLab qual a situação de um certo trabalho em particular.

Por exemplo, num certo projeto designado EXEMPLO.PJT temos duas janelas abertas: exemplo.asm e exemplo.lst.

Em outro projeto designado TESTE.PJT temos três janelas abertas: teste.asm, teste.lst. e File Register Window.

Com o gerenciamento de projetos presente no MpLab não precisamos “lembrar” quais as janelas que cada projeto estava usando no momento em que encerramos o trabalho, pois ao carregarmos o projeto desejado todas as informações relativas ao mesmo serão recuperadas.

**IMPORTANTE:** Ao encerrar seus trabalhos não vá fechando as janelas abertas. Feche apenas o MpLab e na janela “Save Project” selecione **Yes**, ou caso pretenda apenas encerrar o trabalho atual, selecione no menu “Project > Close Project”, e responda **Yes** se necessário, para salvar as alterações.

#### Usando o MpLab pela primeira vez.

Crie uma pasta (diretório) em seu HD com o nome **c:\0pic** (com o número 0 antes, a pasta sempre estará no topo dos menus).

Vamos criar um novo projeto, de nome EXEMPLO.PJT, neste

diretório. Inicie o MpLab. Sua tela deverá ser parecida com a da figura 1.

#### Selecionando o processador

Se o processador indicado no rodapé não for o 16F84A, siga a seguinte seqüência nos *menus*:

*Options* -> *Development Mode*, e selecione a ficha “**Tools**” e teremos a tela da figura 2.

Marque a opção **MPLAB-SIM Simulator** e selecione **PIC16F84A** no *menu* à direita, deixando sua seleção conforme acima e depois dê um *click* no botão **Apply** e em seguida em **OK**.

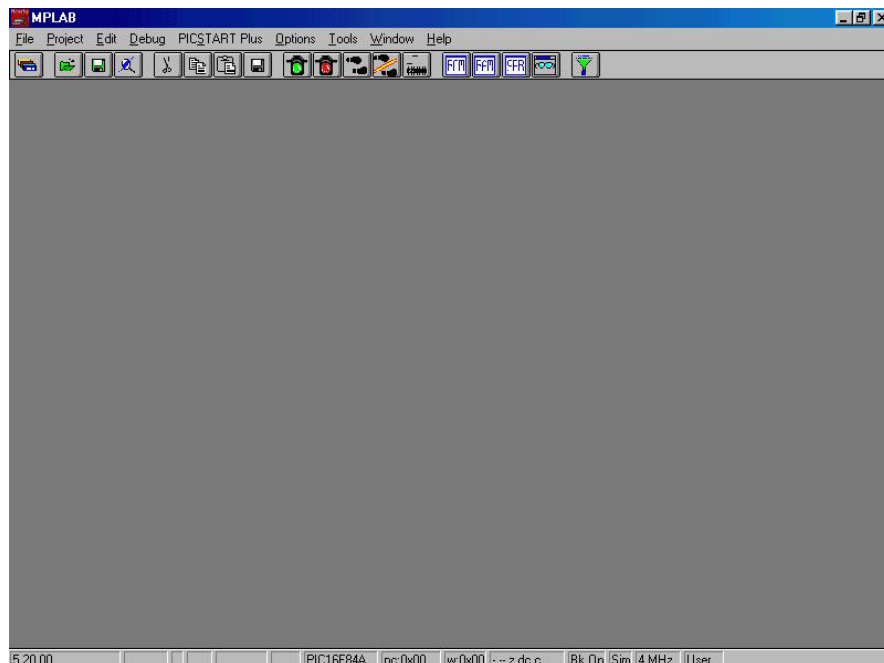


Figura 1 - Tela inicial do MpLab

Sua tela agora deve estar conforme a tela inicial (figura 1).

### Ajustando o *clock* para simulação

Para podermos obter do simulador informações precisas sobre nosso programa quando simulado, devemos informar ao mesmo qual a frequência de *clock* com que o *chip* funcionará. Selecione na linha do *menu* do Mplab

*Options > Development Mode*  
> e selecione a ficha “Clock”  
e obteremos a janela da figura 3:

- Inicialmente selecione se a frequência estará em MHz, kHz ou Hz;
- No campo “Desired Frequency” entre com a frequência desejada, por exemplo, 4 para 4 MHz ( o campo “Actual Frequency” mostra a frequência atual com que o simulador está calculando os eventos);
- No box “Oscillator Type” selecione modo XT;
- Dê um *clique* em “OK” e na nova janela apenas responda “OK” novamente.
- Veja na rodapé do Mplab a frequência sendo reconhecida pelo programa.

## TRABALHANDO COM “PROJETOS”

### Criando um novo fonte para trabalharmos com um projeto

Iniciaremos um **novo** projeto sempre pela criação de um novo fonte. Execute a seguinte seqüência:

*File > New*

Um novo fonte de nome **Untitled1** será criado, e o Mplab lhe perguntará se deseja criar um novo projeto.

Responda “NO” para prosseguir na criação do fonte.

Voce obterá a tela da figura 4:

Ajuste o tamanho da janela **Untitled1** conforme sua conveniência.

Nesta janela digite o texto conforme a figura 5:

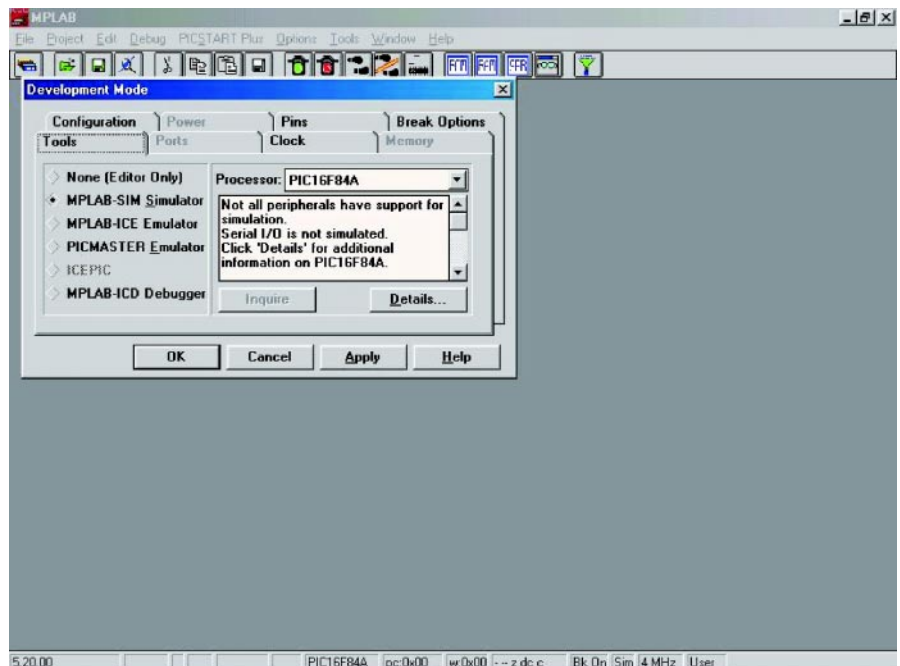


Figura 2 - Seleção do processador.

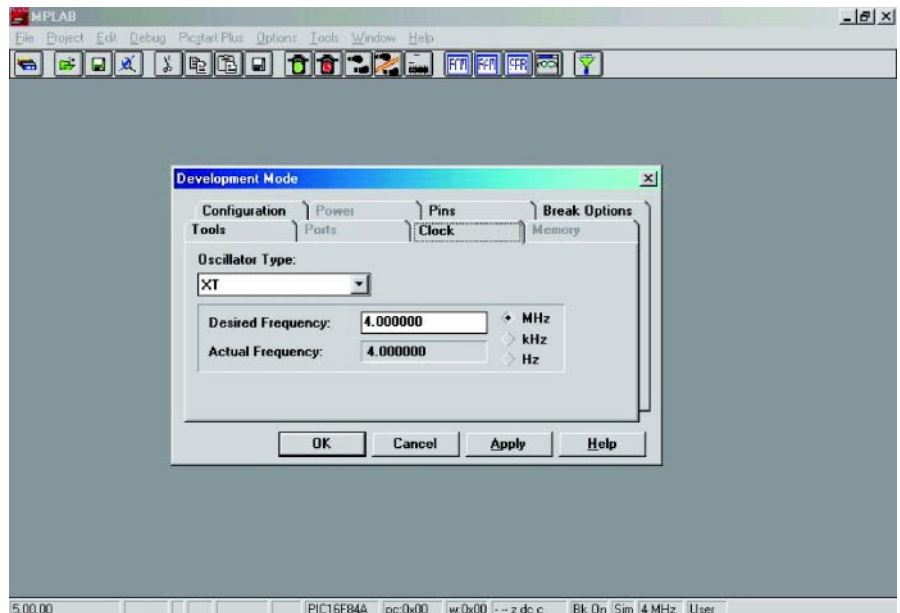


Figura 3 - Ajustando o *clock* para a simulação.

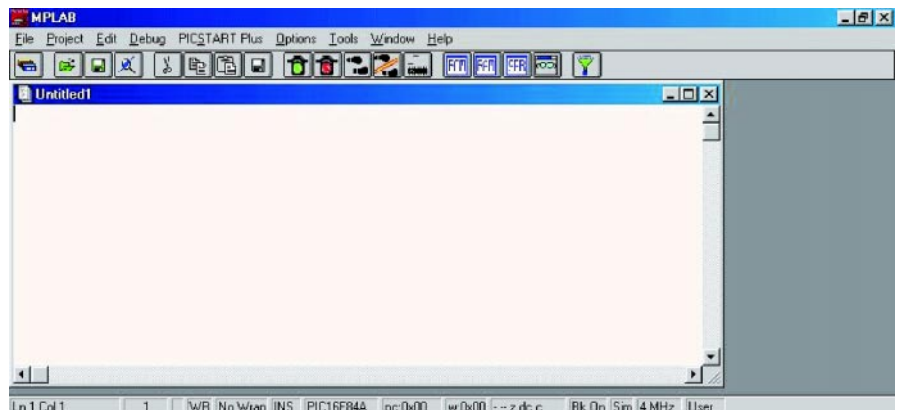


Figura 4 - Janela com o novo fonte ainda sem nome.

Após a digitação (confirme se o seu texto está igual ao acima, sem erros), salve com o nome EXEMPLO.ASM.

**Como?** Após escrever o programa, selecione nos *menus*:

*File > Save as...*

e na janela aberta digite o nome **exemplo.asm** para o fonte E e selecione no drive C a pasta **C:\0pic**.

Agora dê um *click* no botão **OK** e sua tela deverá estar conforme a figura 5, mas com a barra de título do fonte trocada, ao invés de Untitled1, com o nome e caminho completos, c:\0Pic\exemplo.asm.

### Criando o projeto

Agora que já criamos o fonte vamos criar o projeto para trabalharmos com o mesmo.

Selecione **Project > New Project**

e a janela da figura 6 se abrirá (veja na janela do fonte o nome e caminho do arquivo, como dito anteriormente):

Escreva no campo "File Name". **exemplo.pjt** e de um *click* em **OK** (isto é, como salvar na pasta **c:\0Pic** o projeto **exemplo.pjt**)

**IMPORTANTE:** O fonte e o projeto devem estar na mesma pasta

*O nome do fonte não precisa ser o mesmo do projeto.*

Sempre que um novo projeto é criado, a janela **"Edit Project"** será exibida automaticamente.

### Editando o projeto (novo ou já existente)

Entende-se por editar o projeto escolher qual fonte fará parte do mesmo, sendo usado no processo de compilação, além de ajustar os parâmetros do compilador.

Quando o projeto é novo, esta opção aparecerá automaticamente.

Podemos ainda usar a edição do projeto para trocar o fonte ou o compilador usado no desenvolvimento.

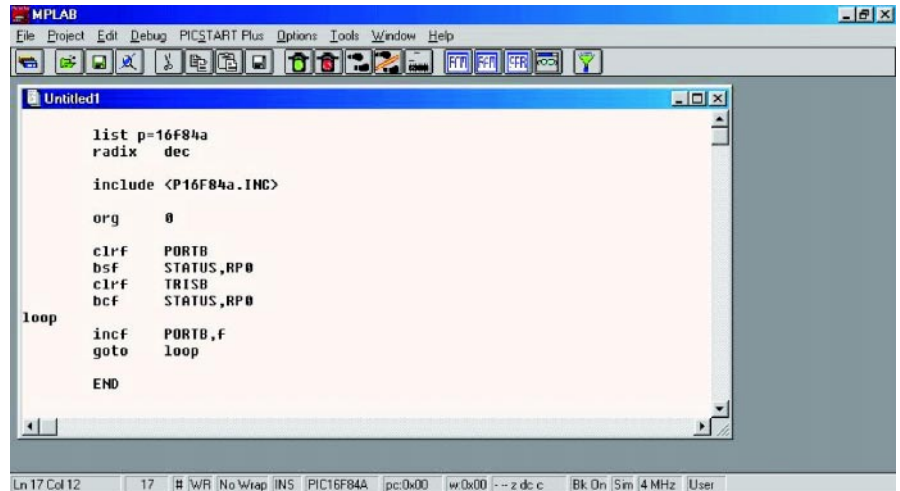


Figura 5 - Fonte de teste já digitado.

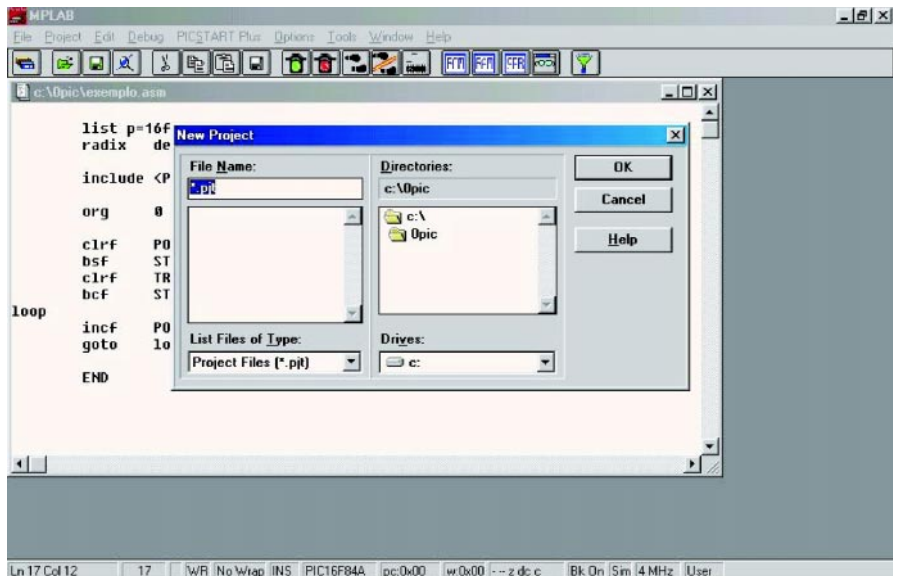


Figura 6 - Janela "New Project" do nosso exemplo.

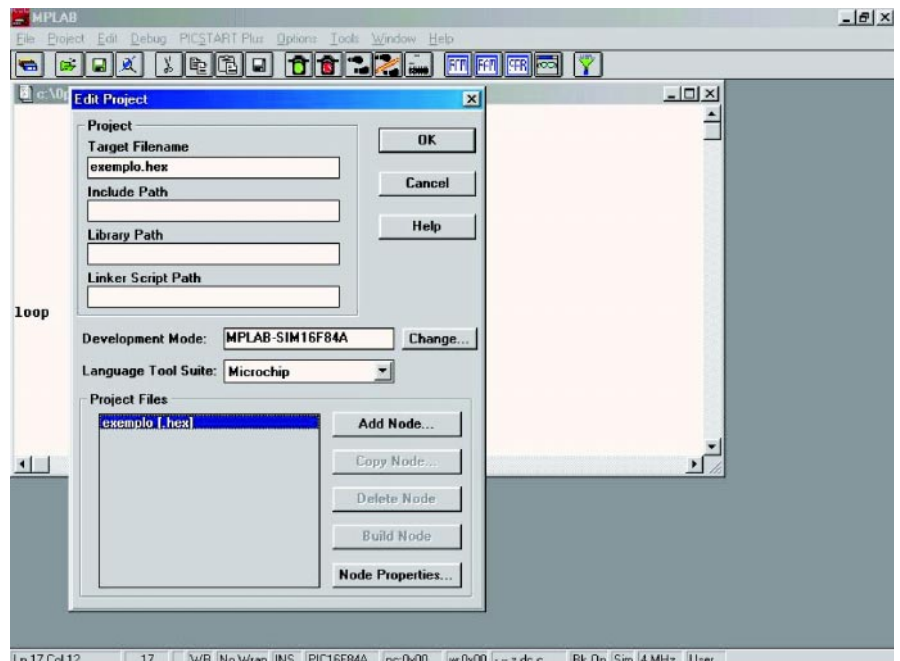


Figura 7 - Editando o projeto exemplo.pjt.



Para indicarmos o fonte de um projeto ou caso desejemos alterar os parâmetros do compilador, devemos usar a sequência:

Project -> Edit Project

e em nosso exemplo teremos a tela da figura 7.

Observe que em nosso caso, o campo “**Target filename**” indica que o alvo, isto é, o objetivo final de nosso trabalho, é a geração do arquivo **exemplo.hex**, que será o arquivo hexadecimal a ser gravado no PIC.

Ajuste o campo “**Development mode**” para exibir **MpLab-SIM, 16F84A**. Ajuste o campo “**Language tool suite**” para **Microchip**.

No Campo “**Project files**” dê um *click* com o *mouse* em cima do texto “**exemplo[.hex]**” e observe que os botões Add Node e Node Properties ficarão realçados (caso ainda não estejam).

Sua janela deve estar como a figura 7 mostrada. Se não estiver, repita os ajustes.

Vamos começar a ajustar o Mplab para trabalhar com nosso “projeto”.

### Ajustando as propriedades do “nó” principal, exemplo.hex

Dê um *click* no botão **Node Properties**. A janela da figura 8 deve

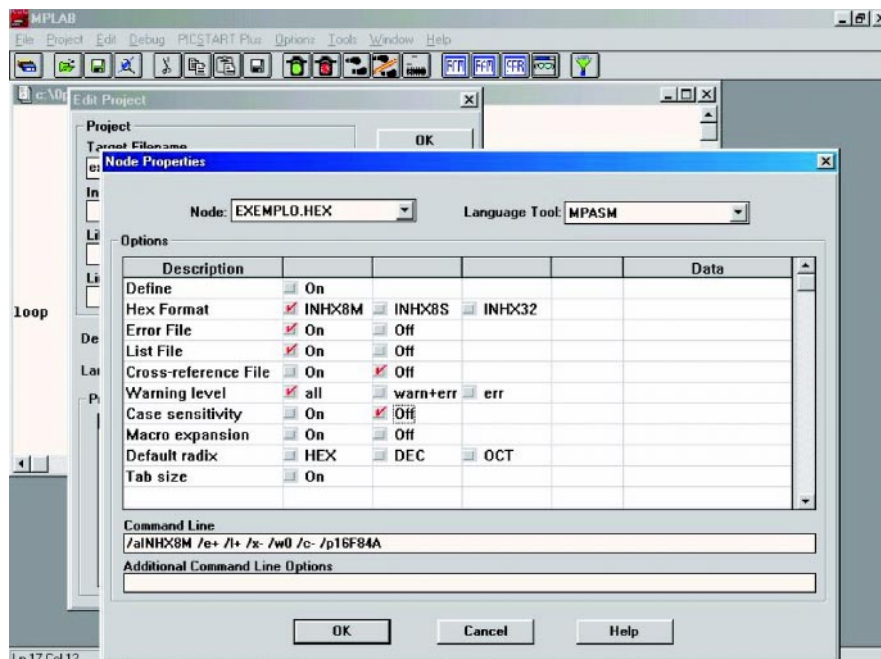


Figura 8 - Janela “Node Properties”.

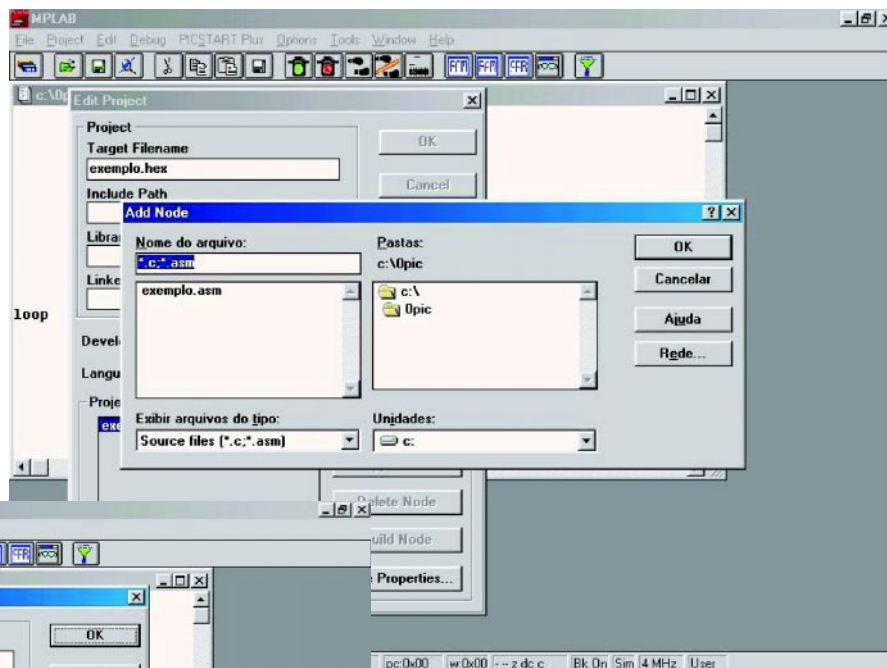


Figura 9 - Janela “Add Node”.

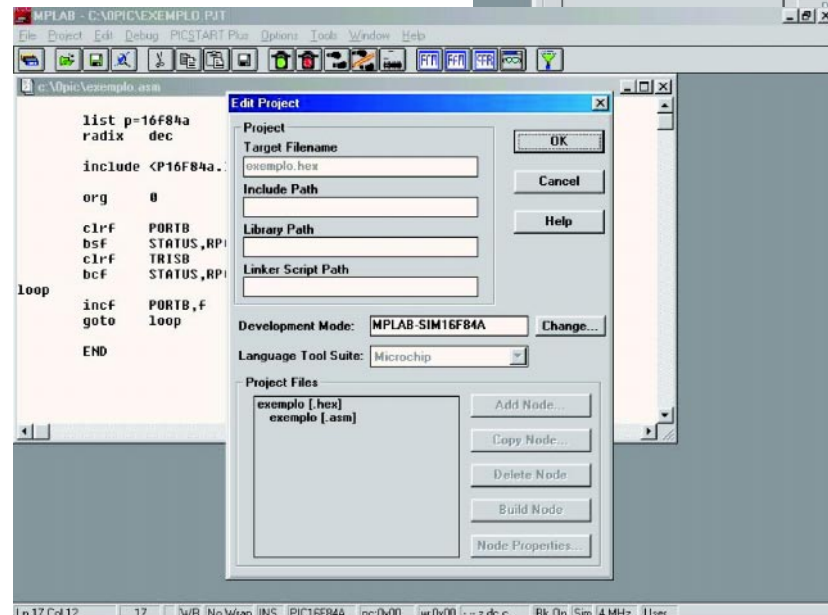


Figura 10 - Janela “Edit Project” completa.

surgir (suas opções pré-selecionadas podem estar diferentes).

Caso as opções marcadas sejam diferentes, use o *mouse* e ajuste os campos de forma que suas opções fiquem iguais às da figura 8 e depois dê um *click* no botão **OK**. **Este ajuste é muito importante !**

Sua tela retornará à da figura 7 (Edit Project).

Neste ponto informamos ao MpLab qual compilador usar, que arquivos gerar (listagem, erros, ...), formato do arquivo

.HEX, entre outros. Para cada projeto novo você deverá ajustar desta forma.

### Adicionando um fonte (também chamado de nó)

Agora vamos informar ao Mplab qual fonte será usado na compilação, dando um *click* no botão **Add Node**. Nossa janela ficará como a figura 9:

Lembre-se: o fonte deve estar na mesma pasta do projeto.

Observe que a janela já aparece com a pasta atual do projeto e mostra em nosso caso o fonte criado anteriormente, exemplo.asm.

**Selecione o fonte e dê um *click* em “OK”.**

Agora sua janela “Edit Project” deve estar conforme a figura 10.

Então, basta dar um *click* no botão **OK** que o processo de edição do projeto estará completo.

Neste ponto você deverá estar com a janela como a da figura 5, mas com os nomes do projeto e do fonte indicados nas barras de título do fonte e do Mplab. A partir deste ponto já podemos compilar e simular o nosso programa.

### OBSERVAÇÕES IMPORTANTES:

Se o Mplab travar repentinamente, todo seu trabalho será perdido.

Como evitar ?

- Selecione **Project > Save Project**
- Selecione **File > Save All**

*Este procedimento de salvar antes de compilar é automatico depois!*

### COMPILANDO O PROGRAMA

Para compilar o fonte basta teclar **F10**. Se não houver erros de sintaxe (escrita) ou outros, haverá a indicação de tudo OK pela janela *Build Results*, mensagem “Build completed successfully”, conforme a figura 11.

Feche a janela *Build Results*.

**IMPORTANTE:** O fato da compilação não apresentar erros não sig-

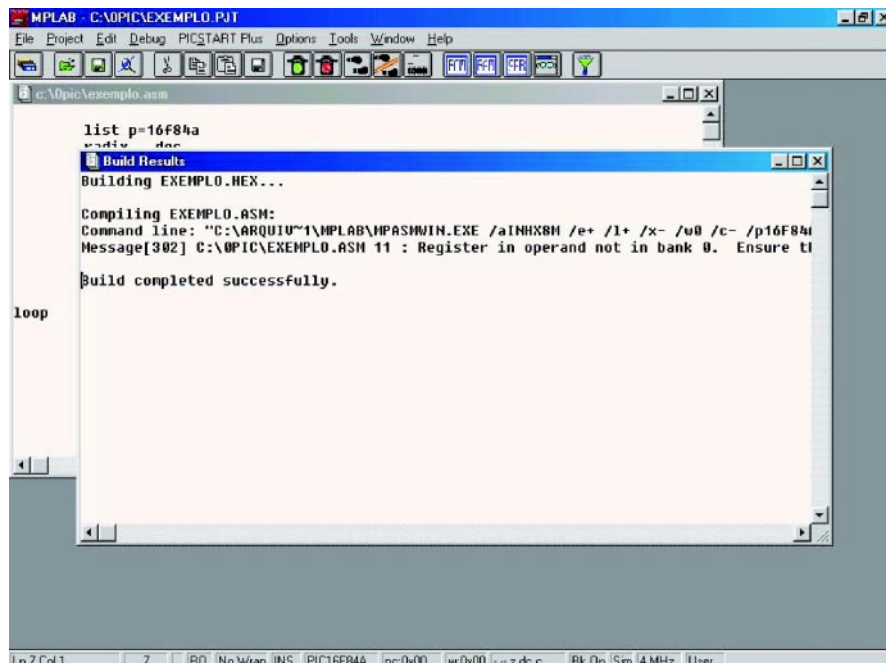


Figura 11 - Janela *Build Results*, após compilação OK.

nifica que o programa funcionará conforme o desejado, pois erros de lógica não são detectados pelo compilador. Se houver erros, uma janela aparecerá indicando todos os avisos e erros detectados pelo compilador.

### Exemplo de erro

Na fonte presente na tela, escreva logo após o org 0, na primeira instrução, PORTAB ao invés de PORTB, e então compile novamente teclando **F10**.

Sua janela *Build Results* deverá ficar conforme a figura 12:

Observe a mensagem ao final, **Build Failed**. Isto é um indicativo de que algo errado no fonte não permitiu a construção do arquivo final.

Observando a janela, vemos uma linha com a seguinte mensagem:

Error[113] C:\0pic\EXEMPLO.ASM 9 : Symbol not previously defined (PORTAB)

Esta mensagem está informando que na linha 9 o símbolo (label, nome, ...) **PORTAB** não está definido.

**IMPORTANTE:** Inicie sempre a análise de erros pelo primeiro erro desde o início.

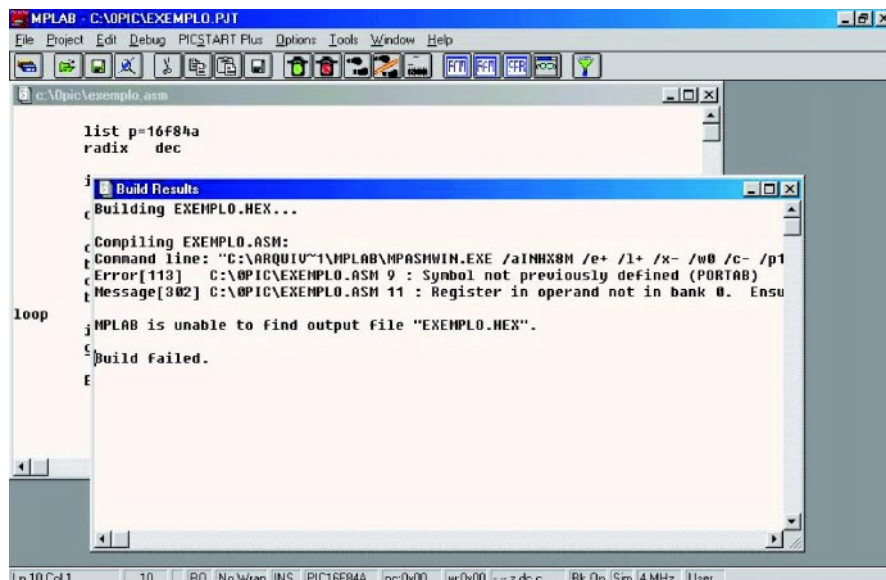


Figura 12 - Janela *Build Results* após erro na compilação.

Onde estão as definições relativas aos registros e *bits* do PIC ?

Selecione

File > View

e abra o arquivo P16F84A.INC na pasta onde instalou o MpLab (Ex.: C:\Arquivos de Programas\Mplab).

Veja que lá estão todas as definições relativas ao PIC 16F84A, como endereços das portas, o nome pelo qual o compilador reconhece os bits, etc. Não esqueça de fechar o arquivo antes de prosseguir.

Onde está o erro ?

Em nosso caso, com apenas algumas linhas de código, fica fácil a localização do erro, mas para fontes maiores (e eles ficarão cada vez maiores), temos outro meio de chegar à linha com o erro.

Vá com o cursor para a linha do primeiro erro, veja a descrição do mesmo e então dê dois *clicks* rápidos com o *mouse* sobre esta linha, e a janela com o fonte será mostrada por cima da janela *Build Results*, com o cursor já posicionado na linha que gerou o erro.

Basta corrigir e repetir todo o processo teclando F10 novamente, até que nenhum erro ocorra.

## Praticando

Repita este exercício de criar erros e identificá-los várias vezes, bem como sair e entrar no MpLab criando outros projetos.

Termine retirando todos os erros e testes realizados, mantendo o projeto exemplo.pjt e seu fonte, exemplo.asm, conforme situação inicial.

## TRABALHANDO COM PROJETOS JÁ EXISTENTES

A principal diferença que encontramos quando abrimos um projeto já existente está no fato de não precisarmos selecionar novamente o fonte ou ajustar o compilador, bastando selecionar nos diretórios existentes o projeto desejado.

Para melhor entendimento, feche o MPLAB e volte para o Windows,

respondendo YES para todas as perguntas, e volte a iniciar o MPLAB.

O MpLab vai perguntar se você deseja abrir o último projeto trabalhado. Se desejar, apenas *click* em **Yes**. Em nosso caso, *click* **No**.

Então selecione o *menu* Project e veja que no rodapé do *menu* que se abriu temos o nome dos últimos projetos em que trabalhamos.

Caso o projeto desejado esteja indicado, basta dar um *click* no mesmo. Veja a figura 13.

Se o projeto desejado não estiver indicado, dê um *click* no item Open Project, e teremos uma nova janela conforme a figura 14.

Escolha o diretório onde está o projeto desejado e depois o nome do

mesmo, e então dê um *click* em OK para ter o projeto aberto.

Se uma janela como a da figura 15 aparecer, apenas *click* em OK e prossiga.

O MpLab está apenas indicando que seu fonte deve ser (re)compilado, pois não achou o hexadecimal do mesmo. Isto pode ocorrer por termos fechado o projeto antes de termos retirado todos os erros, por exemplo.

Neste caso pressione F10 para compilar e prosseguirmos com o exemplo.

A partir deste ponto valem as mesmas regras já vistas sobre edição do projeto, compilação, correção de erros, ...

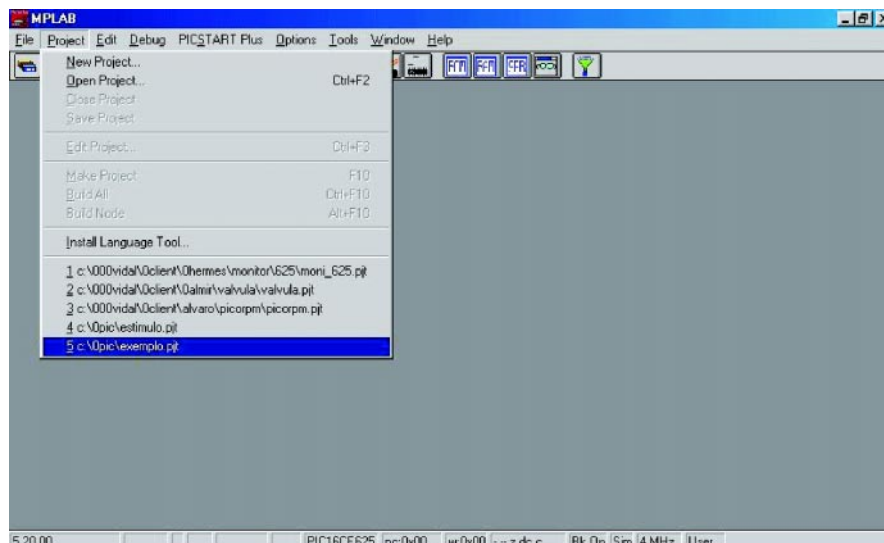


Figura 13 - Menu "Project" com os últimos projetos trabalhados.

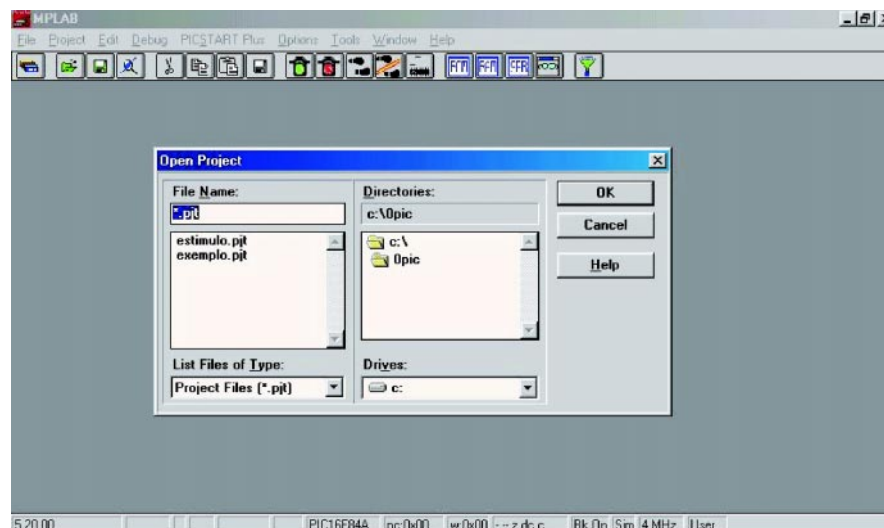


Figura 14 - Selecionando projeto já existente.



## SIMULAÇÃO SIMPLES

Veremos agora técnicas para *simular* nosso programa, desde execução passo a passo até animação.

### Simulando o fonte do projeto EXEMPLO

Para evitar qualquer problema devido a ações erradas que você possa ter realizado, feche o Mplab e abra novamente selecionando o projeto EXEMPLO.PJT

Compile teclando **F10** e corrija eventuais erros de digitação.

### Principais teclas para o controle da simulação

#### F6 RESET

Equivale ao *reset* da CPU.

Posiciona o contador de programa no endereço 0000, e coloca uma barra preta sobre a linha correspondente.

Esta barra indica “a próxima” instrução a ser simulada.

#### F7 STEP

A cada toque em F7 o Mplab executa uma instrução do programa. É como se nosso processador rodasse uma instrução de cada vez. Se for mantida pressionada, executará as instruções no intervalo de repetição automática da tecla.

#### CTRL + F9 ANIMATE

Roda o programa passo a passo dinamicamente, tornando possível acompanhar visualmente a sequência do programa.

#### F9 RUN

Realiza a simulação rápida, sem atualizar a tela. Ideal para simular situações que tomariam demasiado tempo na animação. Apenas a janela **StopWatch** é atualizada (veremos mais tarde)

#### F5 STOP

Interrompe a simulação dinâmica iniciada pelo Ctrl+F9 ou pelo F9

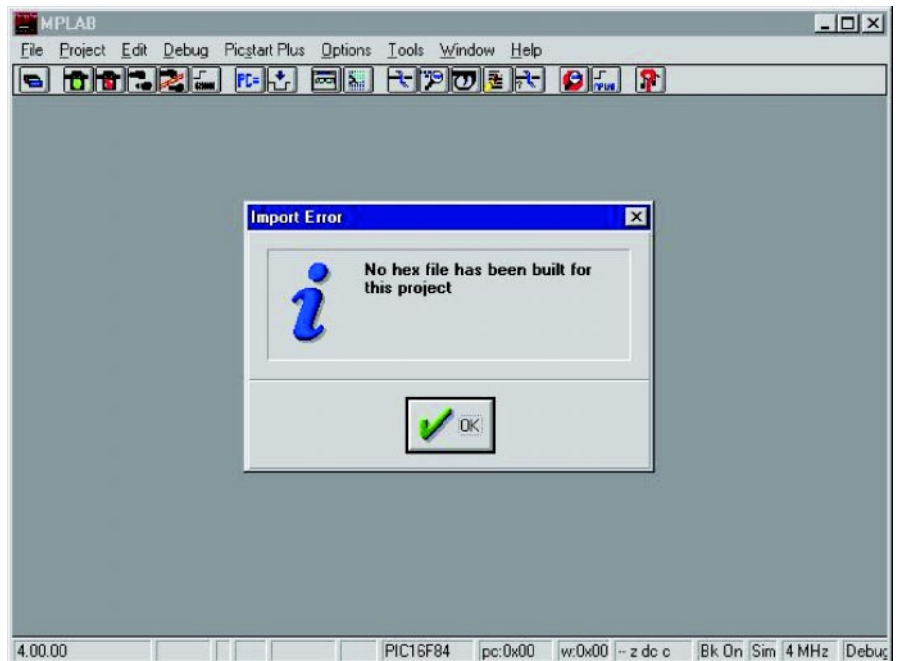


Figura 15 - O projeto aberto ainda não possui um .HEX válido.

### Realizando uma simulação dinâmica para testes

Deixe seu projeto com a aparência igual à da figura 5. Compile (F10) e então pressione a tecla F6. Você deve obter uma tela como a da figura 16. Vá pressionando **F7** e veja que a barra preta, indicativa da próxima instrução a ser simulada, vai se deslocando, acompanhando a execução do programa. Pressione **Control + F9** e veja a simulação ocorrer dina-

micamente. Para parar, pressione **F5**. Veremos a seguir como abrir uma outra janela que nos mostrará o tempo decorrido de simulação.

### Janela Stopwatch. Vendo o tempo de execução

Para abrir a janela *Stopwatch*, siga a sequência no *menu*:

Window > Stopwatch... e ajuste o tamanho e a posição das janelas para que fiquem como a figura 17.

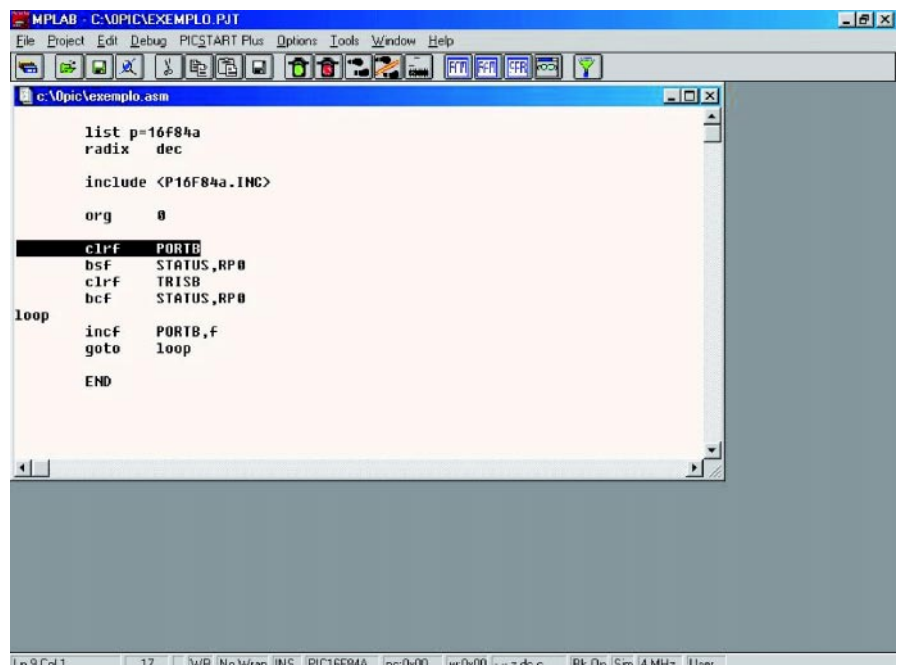


Figura 16 - Tela inicial após o “reset” do processador.

Para que a simulação ocorra, a janela **exemplo.asm** deve ser a janela ativa, como na figura 17 (basta dar um *click* dentro da mesma).

Após qualquer ajuste na janela stopwatch, veja que as teclas de simulação não funcionam, até que a janela do fonte seja selecionada.

Pressione F6 (*reset*) e veja que a cada F7 pressionado os campos “Cycles” e “Time” do stopwatch vão se alterando, indicando o número de ciclos e o tempo decorrido desde o *reset* (ou desde o ultimo *click* no botão *Zero*).

Experimente também com **Control+F9**.

Pare a simulação (F5) e dê um *click* no botão *Zero* do stopwatch, e veja que os valores “Cycles” e “Time” serão zerados, mas ao voltar à janela do fonte, a simulação pode prosseguir do ponto onde parou.

Esta característica permite-nos determinar o tempo exato de um certo trecho de programa.

## Observando registros da CPU durante a simulação

Além de podermos ver o tempo de execução, podemos observar como os registros internos se comportam durante a execução do programa.

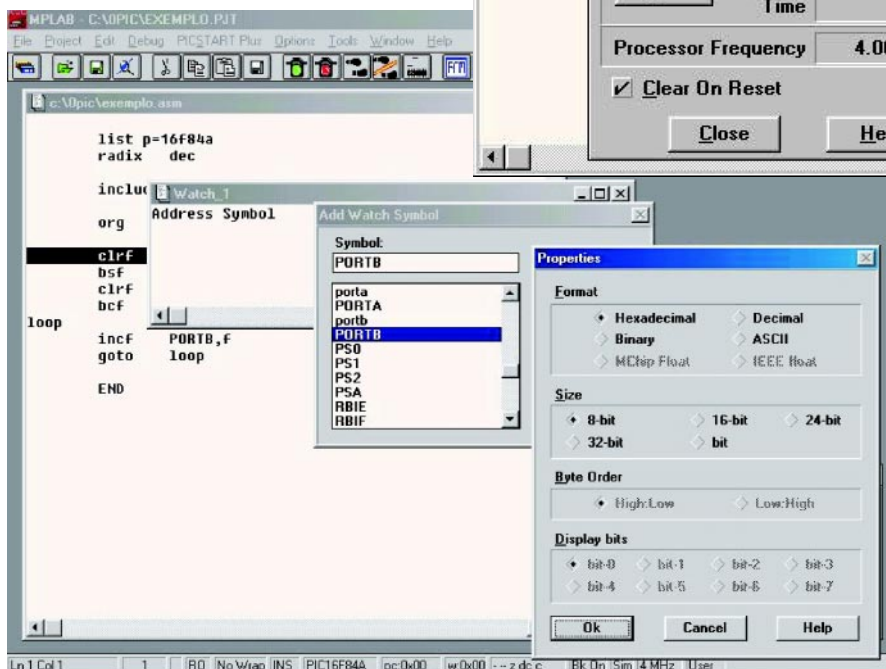


Figura 19 - Ajustando as propriedades do registro escolhido.

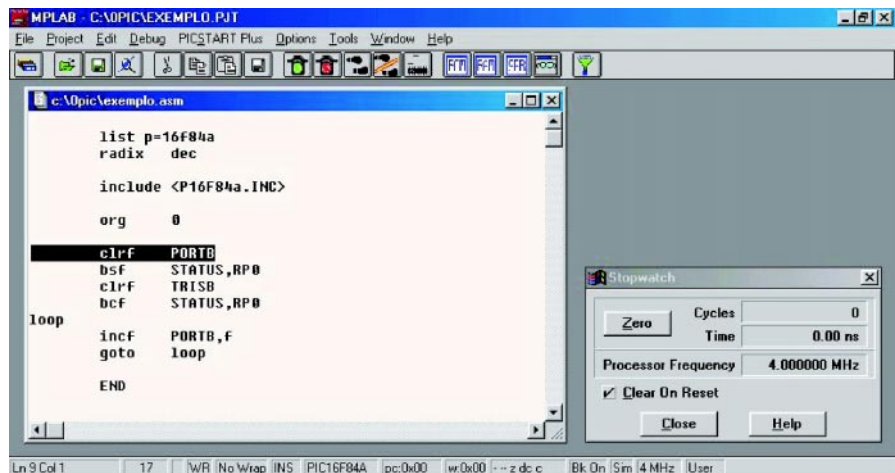


Figura 17 - Janela Stopwatch junto à janela exemplo.asm.

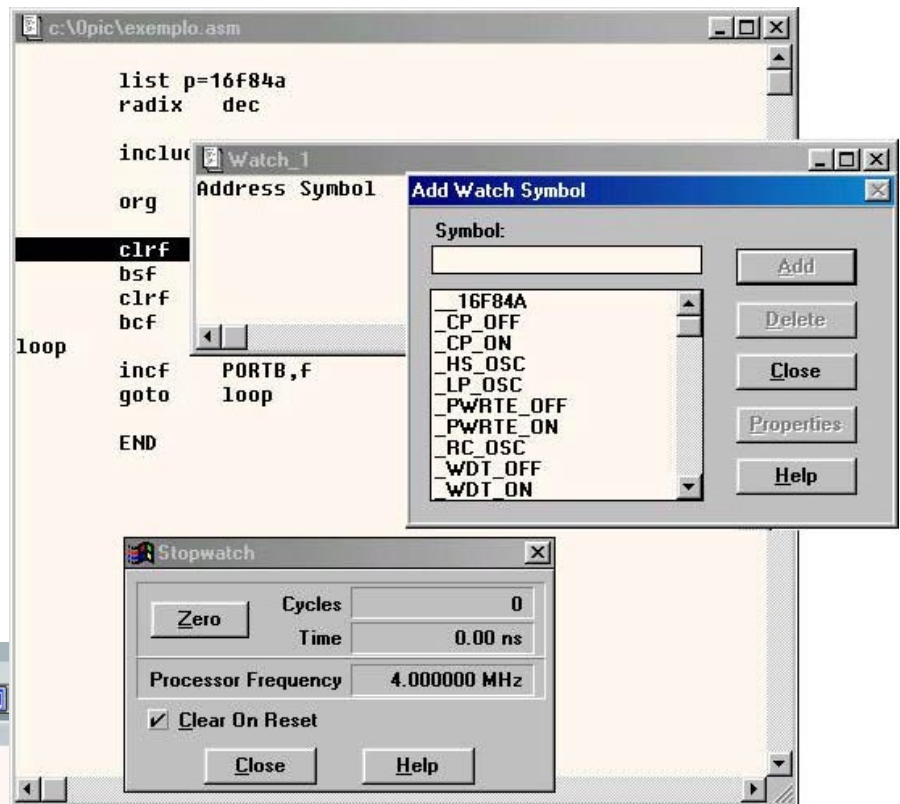


Figura 18 - Iniciando a janela Watch para ver registros.

No *menu* selecione a seqüência: **Window > Watch window > New watch window** e sua janela ficará como a figura 18:

Na janela ativa (**Add Watch Symbol**), no campo “Symbol:” escreva **PORTB** (os registros do PIC devem ser selecionados em maiúsculas) e depois *click* no botão **Properties**. Você pode ainda correr a barra de rolagem vertical e selecionar com o *mouse* o registro desejado. Sua tela deverá estar como a da figura 19, com uma nova janela



onde ajustaremos as “propriedades” de visualização do registro selecionado.

Ajuste *Format* para DECIMAL e *Size* para 8 bits, e então dê um *click* no botão **OK**, fechando a janela de propriedades do registro selecionado.

Veja que ao fechar a janela **PROPERTIES** a janela **Watch\_1** já conterá a seguinte linha:

Address	Symbol	Value
06	PORTB D	'0'

**IMPORTANTE:** O valor inicial do PORTB aqui indicado pode ser diferente do seu!

Para ficar mais fácil acompanhar a simulação ajuste o tamanho e a posição da janela Watch\_1 para que sua tela fique conforme a figura 20.

Se desejar, acrescente outros registros da CPU e então dê um *clique* em “Close” para fechar a janela “Add Watch Symbols”. Faça da janela exemplo.asm a janela ativa e divirta-se com os recursos de simulação do MpLab.

Com **Control + F9** veja o programa rodando, o tempo decorrido e a variação do PORTB.

Se precisar de maiores detalhes sobre as possibilidades de simulação, consulte o Help do MpLab, em

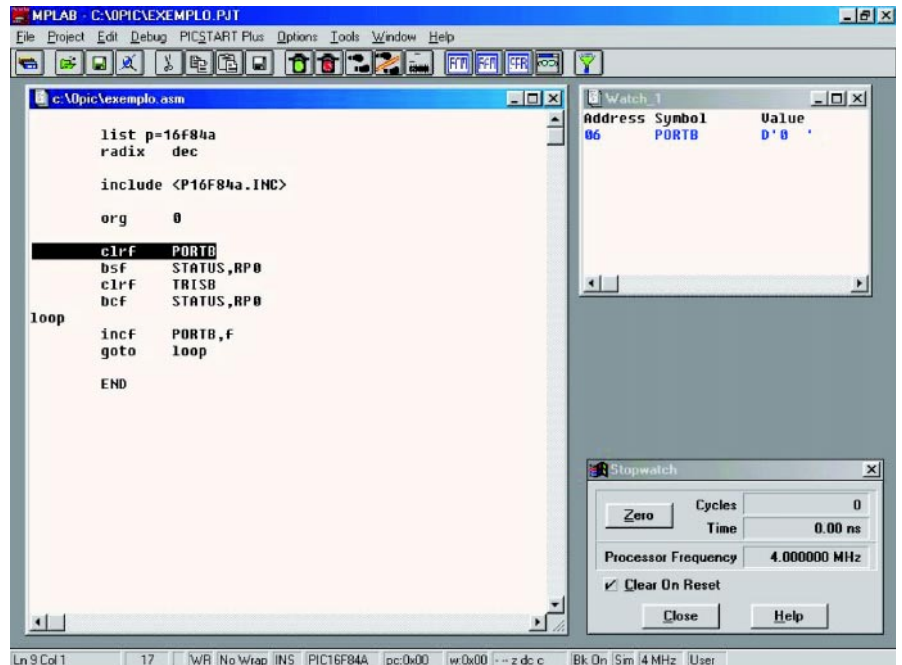


Figura 20 - MpLab com fonte, stopwatch e visualização do PORTB.

Help > MPLAB Help

## SIMULANDO SINAIS EXTERNOS NO MPLAB

### Objetivo

Demonstrar a capacidade do MpLab em simular variações externas nos pinos do PIC.

### Procedimento

Inicie o MpLab e certifique-se de

que o mesmo está no modo de simulação do 16F84A.

Seguindo os passos já vistos para a criação de fonte, projeto, dimensionamento de janelas e simulação, crie um novo fonte e um novo projeto na pasta c:\0pic de nomes ESTIMULO.ASM e ESTIMULO.PJT.

Na janela Watch\_1 acrescente o PORTA com propriedade em binário e o PORTB com propriedade em decimal.

Neste exemplo o PORTB será incrementado somente se RA0 estiver em 1.

No fonte, conforme já visto anteriormente, digite (respeitando as tabulações):

```
list    p=16F84A
include <P16F84A.INC>
radix  dec

org     0

clrf    PORTB    ; inicia em 0
bsf     STATUS,RP0    ; banco 1
                    ; para ajuste do trisb

clrf    TRISB    ; portb é toda saída
bcf     STATUS,RP0    ; banco 0
loop:
    btfss PORTA,0    ; se RA0 = 1 pula
                    ; 1 linha
```

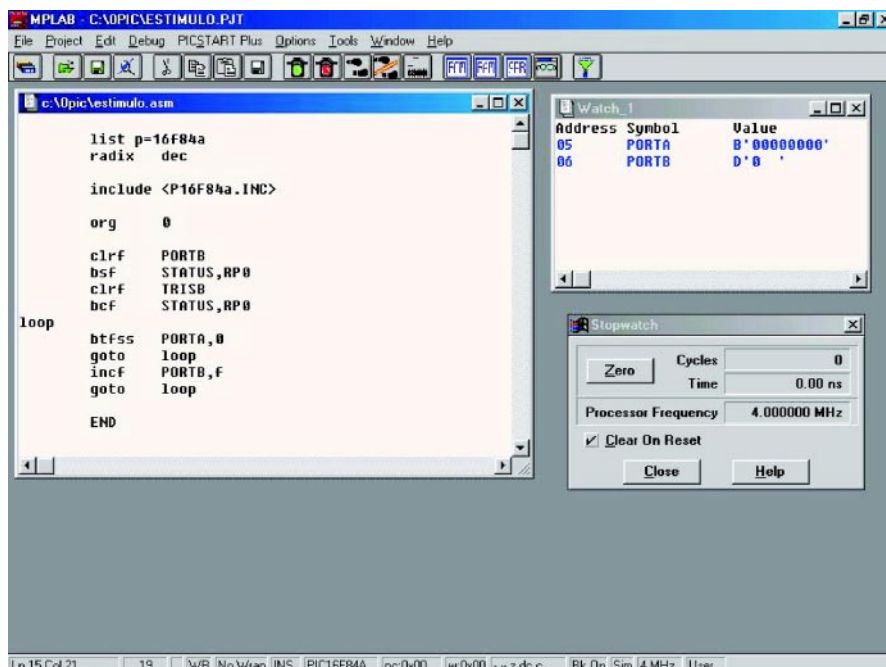
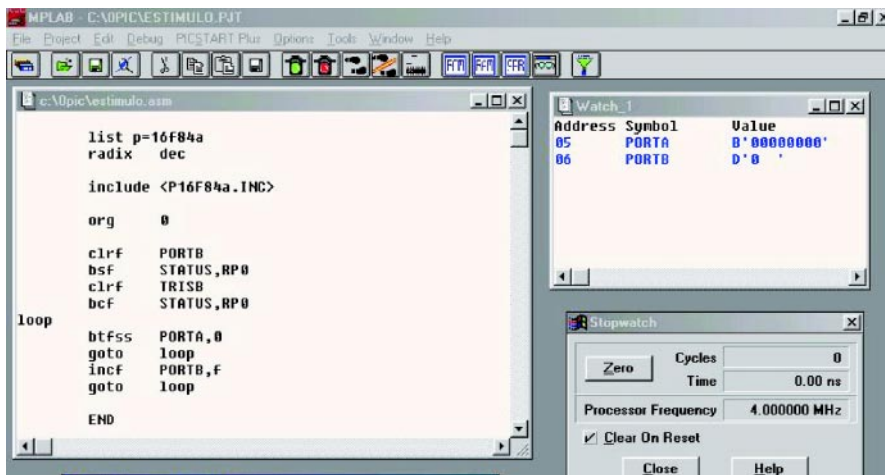


Figura 21 - Projeto e Fonte ESTIMULO prontos.



**Low (L)** Coloca nível '0' no pino

**High (H)** Coloca nível '1' no pino

**Toggle(T)** Inverte o estado no pino (inicia sempre em 0)

Experimente dar uma função diferente para cada botão e observa o resultado.



Figura 22 - Botões de estímulo.

```
goto loop ; RA0 = 0, repete
; o teste
incf PORTB,f ; se chegou
; aqui, RA0 = 1
goto loop ; repete tudo
END
```

Sua tela deverá estar como a figura 21.

No menu principal selecione:

Debug > Simulator Stimulus > Asynchronous Stimulus e você deverá obter uma janela conforme figura 22.

Observe que existem 12 “botões” com nomes de Stim 1 a Stim 12 e dentro dos parênteses a letra P.

Vamos ver cada ajuste em detalhes.

Com o botão direito do mouse dê um click no botão **Stim 1 (P)** e você obterá o menu da figura 23.

Observe que a opção **Pulse** está selecionada. Esta é a opção padrão, indicada dentro dos parênteses como (P).

As opções para estímulo externo são:

**Pulse (P)** Dá um pulso no pino selecionado (de 0 para 1 e retorna a 0)

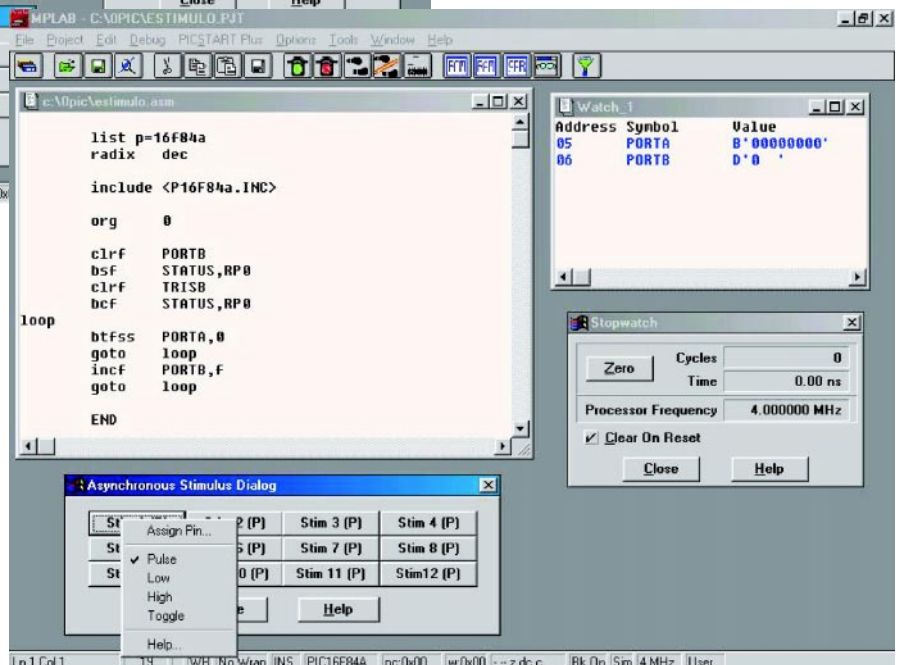


Figura 23 - Ajustando os botões de estímulo.

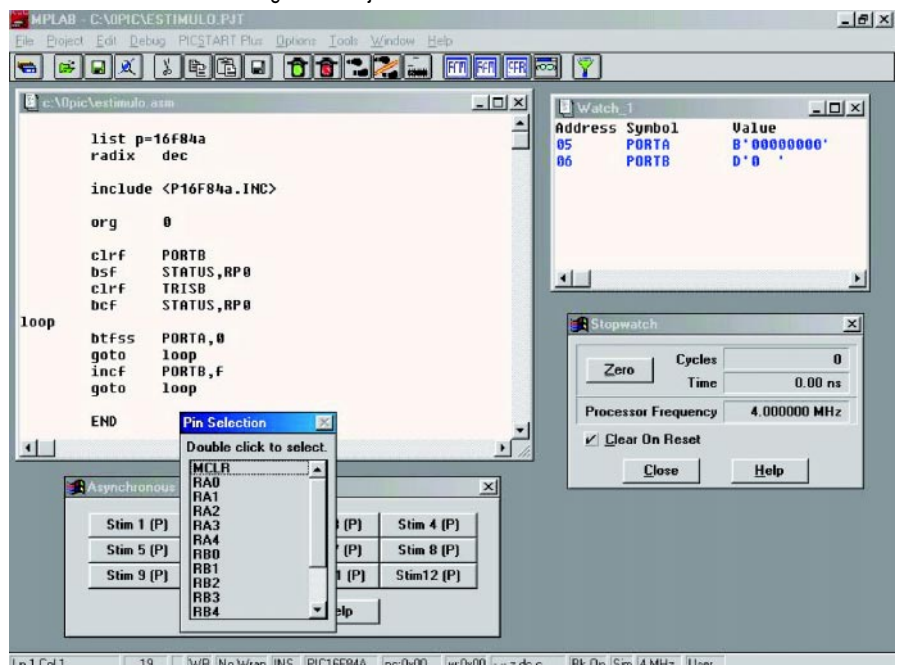


Figura 24 - Ajustando os pinos que receberão o estímulo.

Como relacionar os pinos do PIC aos botões

Observe que a primeira opção do menu é **Assign pin....**

Selecionando esta opção um novo menu se abre, conforme a figura 24.

Então, para associarmos o botão desejado a um certo pino do PIC, basta dar um duplo *click* no pino desejado.

Como treinamento, faça os seguintes ajustes:

Botão 1: RA0 (T)

Botão 2: RA1 (L)

Botão 3: RA2 (P)

e veja se os botões ficaram conforme a figura 25.

Agora, durante a simulação, podemos fazer com que sinais externos sejam aplicados ao PIC e ver seu comportamento.

Basta durante a simulação dinâmica (**CTRL F9**) *clique* sobre os botões desejados, e o estímulo associado ao mesmo será aplicado ao PIC e o simulador responderá de acordo.

Exemplo com estímulo

Baseado no projeto ESTIMULO.PJT já pronto, vamos

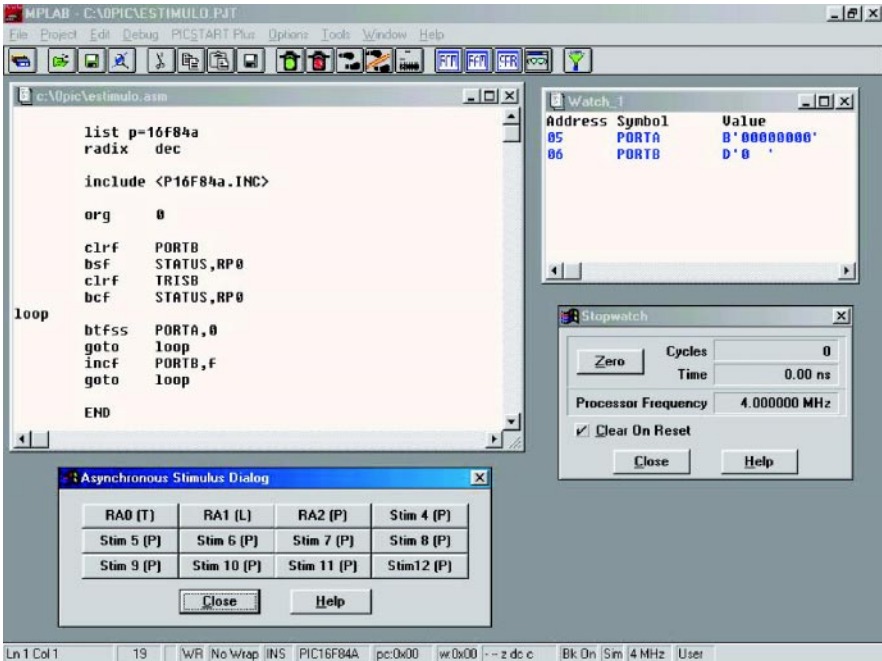


Figura 25 - Botões de estímulo já ajustados.

efetuar a simulação com estímulo no pino RA0.

Certifique-se que temos o seguinte ajuste: **RA0 (T)**

Após a compilação e eventuais correções, desloque a janela de estímulo para um ponto que não atrapalhe a visualização dos registros e faça do fonte a janela ativa.

Tecla **CTRL F9** e veja na simulação que a instrução *btfss porta,0* inicialmente sempre vê a entrada RA0 em 0.

Dê vários *clicks* bem espaçados no botão RA0 (T) e vá analisando o comportamento do PIC.

Boa Sorte  
Vidal





