

MPLAB IDE

Instruções básicas de utilização

Carlos Roberto da Silveira Jr

Instalação

O MPLAB IDE é o software produzido pela fabricante de microcontroladores Microchip para, podendo ser adquirido gratuitamente em [HTTP://www.microchip.com](http://www.microchip.com). Este programa permite edição, compilação, simulação de programas em assembler e linguagem C para microcontroladores da família PIC Microchip.

Caso tenha dificuldades em obter o arquivo pelo site procure-me que poderei passá-lo.

O processo de instalação é simples, não havendo necessidades de maiores informações.

Edição

O processo de edição pode ser feito abrindo um projeto base ou criando um novo projeto a partir do zero. Ambos projetos devem ter um caminho menor do que 62 caracteres, uma limitação do MPLAB que pode apresentar erros no processo de compilação do projeto. Apresenta-se esse possível problema aqui para que o usuário possa organizar nomes e locais de arquivos levando em conta essa limitação. Assim programas salvos em locais como Área de Trabalho ou em Meus Documentos podem apresentar como erro de compilação essa limitação do próprio MPLAB.

Criando um projeto do zero

Após a instalação do MPLAB clique no link que foi criado na área de trabalho, se não tiver o link, vá em Iniciar → Todos os Programas → Microchip → MPLAB IDE v7.50 → MPLAB IDE. O programa abre conforme a figura 1.

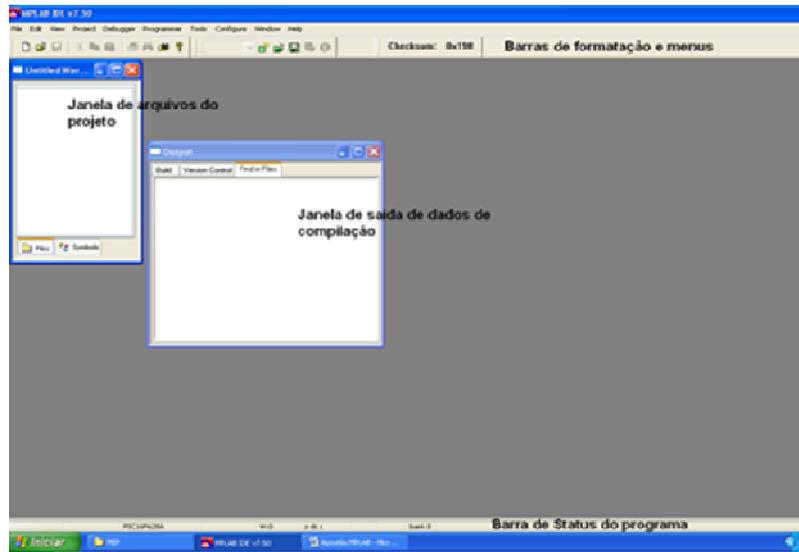


FIGURA 1. JANELA INICIAL DO MPLAB.

A janela de arquivos do projeto apresentará todos os arquivos relacionados ao projet. Para criar um novo projeto vá em Project → New, abrirá uma janela para inserir o nome e pasta do projeto, como visto na Figura 2.

Crie o projeto com o nome PISCALÉD A e pasta de destino C:\MP\PISCA. A pasta de destino deve ser criada. Caso ela não esteja criada no momento o MPLAB perguntará se quer criá-la.

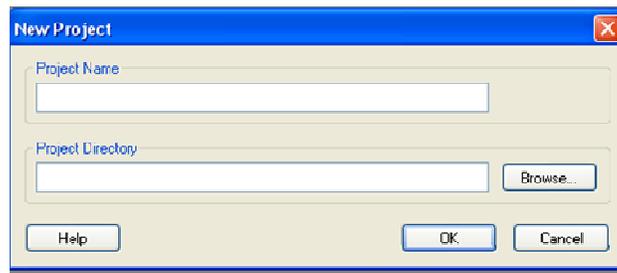


FIGURA 2. JANELA PARA CRIAÇÃO DE NOVO PROJETO.

Após o projeto criado ve-se que a janela do projeto apresenta diversas pastas, como visto na Figura 3.

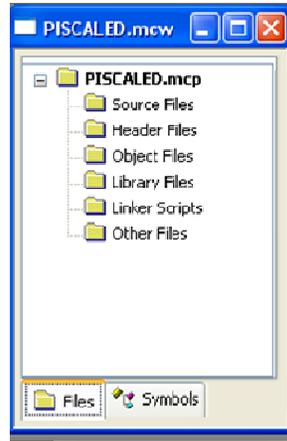


FIGURA 3. JANELA DO PROJETO PISCALED.

Observa-se que existem vários tipos de arquivos, dentre eles tem-se:

- Source Files (Arquivos Fonte), composto por arquivos de programas feitos em assembler, linguagem C.
- Header Files (Arquivos Cabeçalhos), composto por arquivos de especificações do microcontrolador.
- Object Files (Arquivos Objeto), composto por arquivos objetos, arquivos pré-compilados, muitas vezes são módulos que são utilizados em vários projetos (como rotina de display LCD, sensors, etc). São utilizados para ter-se um processo de compilação mais rápido.
- Library Files (Arquivos Biblioteca), composto por bibliotecas de sub-rotinas que são chamadas pelo arquivo fonte.

Configurações Iniciais

Após a criação do projeto é necessário fazer algumas configurações iniciais. Para selecionar o debugador, ou simulador do programa. O debugador padrão do MPLAB é o MPLABSIM selecionado em Debugger → Select Tool → MPLAB SIM. Para selecionar o microcontrolador vá em Configure → Select Device. A Figura 4 apresenta a janela, observa-se que, conforme o microcontrolador escolhido, tem-se as informações de que suporte o MPLAB fornece.

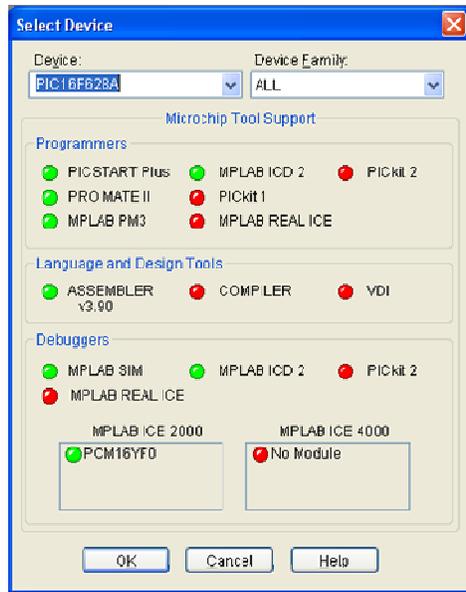


FIGURA 4. SELEÇÃO DO MICROCONTROLADOR.

A palavra de configuração corresponde às configurações que devem ser conhecidas pelo microcontrolador antes mesmo do início de execução do programa do interno. A gravação é feita em uma palavra de 32 bits no endereço 2007H. O valor da palavra muda conforme modificações feitas em cada uma das configurações. Abaixo são apresentadas as opções de cada configuração.

- **Tipo de Oscilador**
 - RC_CLKOUT → Externo, RC, pino 15 CLKOUT, ¼ Frequência
 - RC_I/O → Externo, RC, pino 15 RA6
 - INTOSC_CLKOUT → Interno, RC, pino 15 CLKOUT, ¼ Frequência
 - INTOSC_I/O → Interno, RC, pino 15 RA6
 - EC_I/O → Externo, Circuito, pino 15 RA6
 - LP → Externo, Cristal Ressonador (abaixo de 200KHz)
 - XT → Externo, Cristal ou Ressonador (até 4MHz)
 - HS → Externo, Cristal ou Ressonador (maior que 4MHz)
- **Power Up Timer**
 - Se ativo o PIC só inicia após 72 ms de alimentação
- **Brown Out Detect**
 - Funciona como um sistema de detecção automática de baixa tensão, capaz de resetar o PIC, quando a alimentação menor que 4V por mais de 100us
 - Para outras especificações é necessário um circuito externo
- **WatchDog Timer**
 - Se ativo deve-se lembrar de limpar o contador do WDT periodicamente
- **Master Clear Enable**
 - Se habilitado o pino 4 funciona como MCLR e reseta o microcontrolador quando em 0
- **Low Voltage Program**

- Se habilitado o pino 10 funciona como PGM, permite gravação em 5V (não mais em 13V). Assim um PIC pode gravar em outro, fazer atualização de firmware remotamente
- Data EE Read Protect
 - Se habilitado não permite leitura dos dados da EEPROM interna
- Código de Proteção
 - Impede leitura do firmware do microcontrolador, não impede futuras gravações

No caso deste projeto utilizaremos a configuração mostrada na Figura 5, sendo o valor gravado no endereço da palavra de configuração igual a 3F38H. Ou seja, utilizaremos o oscilador interno de 4MHz e o Master Clear Reset, ou MCLR, acionado pelo seu respectivo pino. Todas as outras funcionalidades estarão desabilitadas.

Address	Value	Category	Setting
2007	3F38	Oscillator	INTOSC: I/O on RA6/OSC2/CLKOUT, I/O on RA7/OSC1/CLKIN
		Watchdog Timer	Off
		Power Up Timer	Disabled
		Brown Out Detect	Disabled
		Master Clear Enable	Enabled
		Low Voltage Program	Disabled
		Data EE Read Protect	Disabled
		Code Protect	Off

FIGURA 5. PALAVRA DE CONFIGURAÇÃO.

Configuração de simulação, como velocidade do oscilador, podem ser configuradas indo em Debugger → Settings... Abrindo a janela Simulator Settings que possui várias abas, como visto na Figura 6.

- Na aba Osc/Trace é possível configurar a velocidade do oscilador.
- Em Break Options configura-se retorno de erros de execução além do tempo do WatchDog.
- Em SCL configura-se opções de retorno de erro e avisos de SCL.
- Em Limitations verifica-se as limitações de simulação para o microcontrolador utilizado.
- Em UART1 IO configure-se a simulação da porta serial.
- Em Animation/ Real Time Updates configure-se os tempos das opções de simulação.

Pela Figura observa-se que o PIC16F628A possui limitações para a simulação da comunicação serial, além de alguns outros periféricos.

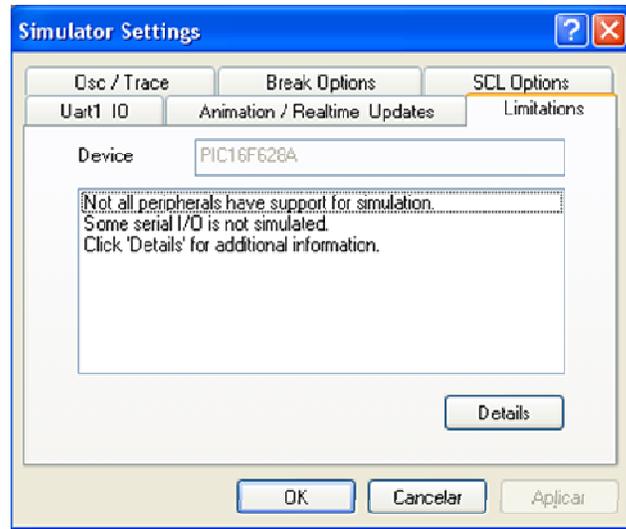


FIGURA 6. CONFIGURAÇÕES DO SIMULADOR.

Criando um Arquivo Fonte

Para criar um arquivo fonte vá em File → New, abrirá um novo arquivo que poderá ser salvo no formato desejado. Copie para o arquivo o código do programa que está no Anexo 1. Salve o programa como PISCALED. O programa, quando executado no Kit Didático da disciplina, apaga todos os LEDs, pisca o LED Amarelo e aguarda o pressionamento da chave 1. Caso seja pressionada pisca o LED Vermelho.

Ao colar observa-se que o texto está em cor de fonte preta. Agora para salvar o arquivo vá em File → Save as... e salve-o na respectiva pasta (C:\MP\PISCA\) com o nome PISCALED.ASM. Após salvar o texto fica colorido pois representa um arquivo fonte. Sendo o padrão verde para comentário (linha iniciada com ";"), azul para instruções e roxo para referências e variáveis.

Para inserir um arquivo fonte ao projeto é necessário ir á janela do projeto e clicar com o botão direito sobre a pasta de arquivos fonte (Source Files), posteriormente clicar na opção Add Files... Abrirá uma janela onde deve-se apontar ao arquivo que a ser inserido, neste caso PISCALED.ASM.

Após esse passo é possível ver que o arquivo PISCALED.asm agora faz parte do projeto, inserido na pasta arquivos fonte.

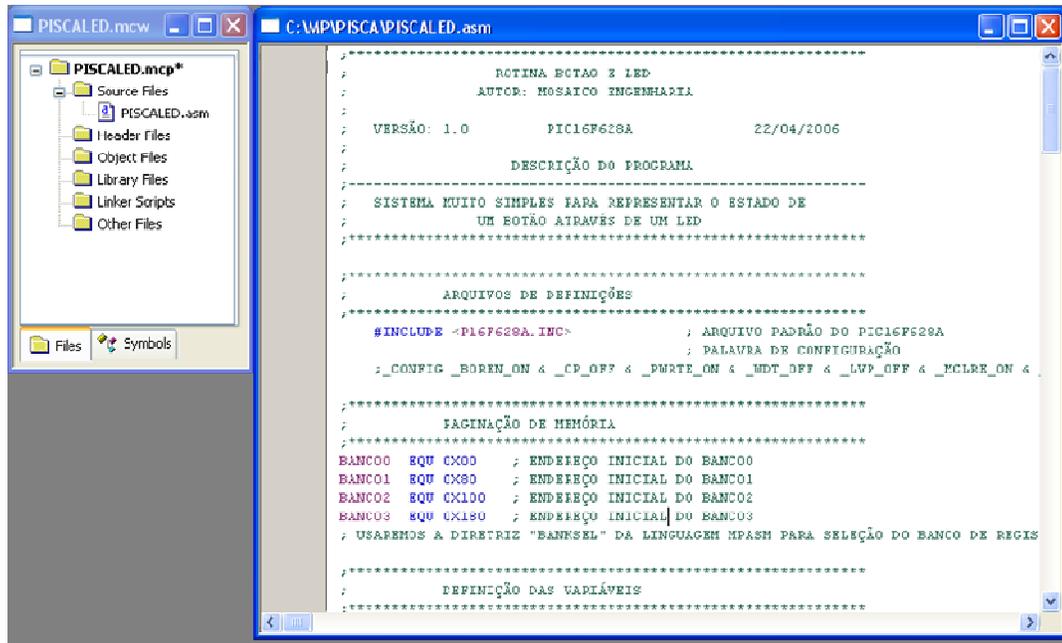


FIGURA 7. TELA DO PROJETO.

Simulando o Projeto

Para similar é necessário inicialmente compilar o projeto, podendo ser feito de várias formas: vá em Project → Build All, CTRL+F10 ou o botão Build All na aba Project Manager. Caso o programa não apresente erros de compilação o MPLAB apresentará a janela de compilação com barra verde marcando 100% compilado sem erros. A janela Output apresentará os resultados da compilação, como visto na Figura 8. Neste caso a compilação não apresentou erros. A seta verde apontando para a primeira linha de código indica que a simulação pode ser iniciada.

Nesse processo alguns arquivos são criados, utilizados para geração do arquivo hexadecimal, e posteriormente são apagados, como visto na janela Output. Algumas mensagens podem ser geradas mas talvez não comprometam a execução do programa. Deve-se analisar as mensagens e suas consequências. Os erros quando gerados não permitem a continuação da compilação do projeto e deve-se analisar o erro gerado através do clique sobre a mensagem de erro, na janela de Output.

Na Figura 9 é apresentado o mesmo código com um erro de compilação. Ao ver o erro na janela Output pode-se clicar sobre a linha explicativa do erro e o MPLAB apontará para a respectiva linha de erro no código do projeto. Neste caso pode-se ver que separou-se as letras da instrução.

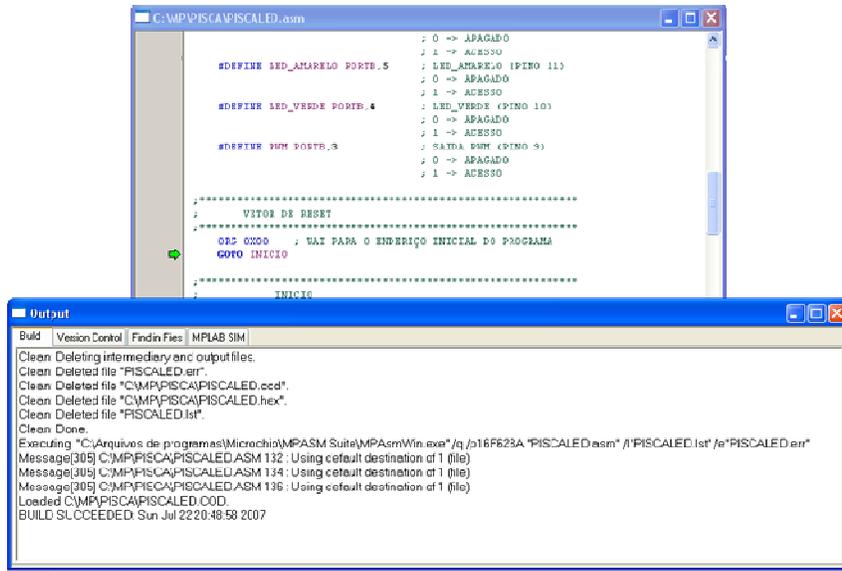


FIGURA 8. JANELA OUTPUT APÓS A COMPILAÇÃO.

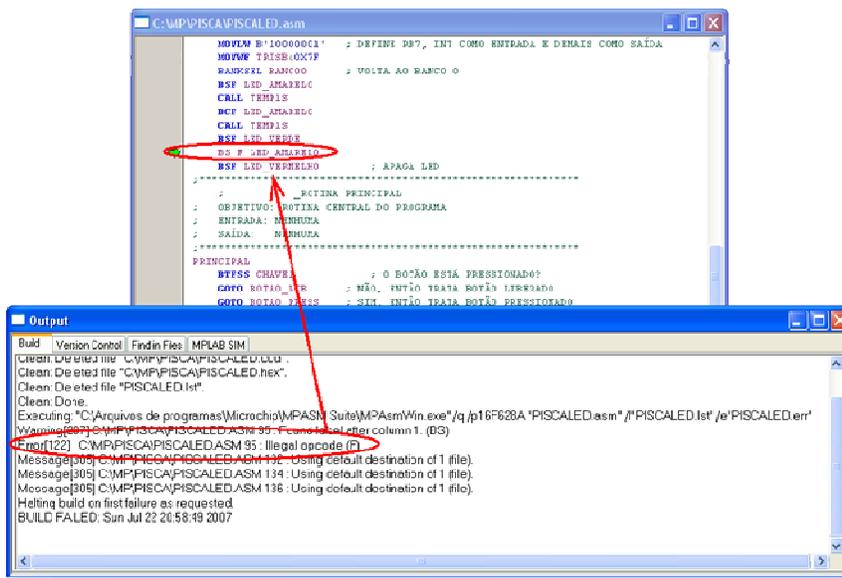
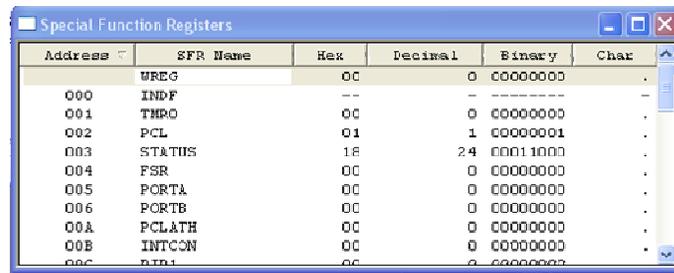


FIGURA 9. EXEMPLO DE COMPILAÇÃO COM ERRO.

Após a confirmação de que os erros foram sanados e as mensagens não ocasionam problemas na execução do programa, pode-se iniciar a simulação.

Janelas de Simulação do Projeto

Algumas janelas auxiliam na simulação do programa. As janelas de registradores especiais permite visualizar todos os registradores de uso especial do respectivo microcontrolador, acessada por View → Special Function Registers.



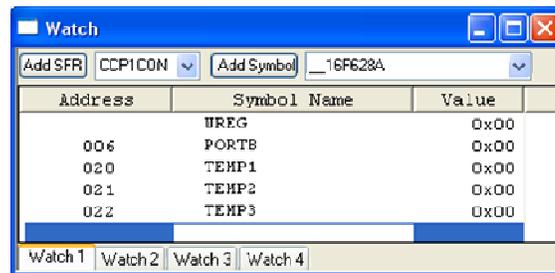
Address	SFR Name	Hex	Decimal	Binary	Char
000	WREG	0C	0	00000000	.
001	INDF	--	--	-----	.
001	TMPO	0C	0	00000000	.
002	PCL	01	1	00000001	.
003	STATUS	1E	24	00011000	.
004	FSR	0C	0	00000000	.
005	PORTA	0C	0	00000000	.
006	PORTB	0C	0	00000000	.
00A	PCLATH	0C	0	00000000	.
00B	INTCON	0C	0	00000000	.
00C	TRISA	0C	0	00000000	.

FIGURA 9. JANELA DE REGISTRADORES DE FUNÇÕES ESPECIAIS.

A janela Watch permite visualizar variáveis e registradores que necessitam ser acompanhados durante a execução do programa. Acessada através de View → Watch.

Para inserir um registrador clique sobre a linha em branco, no campo Symbol Name e escreva o nome do registrador. Para selecionar um registrador especial pode-se selecionar e depois clicar no botão Add SFR.

Para mudar o formato do dado clique com o botão direito sobre o campo valor da variável e selecione Propertiers..., abre-se uma janela onde pode-se modificar o formato.



Address	Symbol Name	Value
	WREG	0x00
006	PORTB	0x00
020	TEMP1	0x00
021	TEMP2	0x00
022	TEMP3	0x00

FIGURA 10. JANELA WATCH.

A janela StopWatch permite registrar o tempo de execução de uma rotina ou tempo para chamada de uma interrupção. Para utilizá-la vá em Debugger → StopWatch.

O botão Zero permite inicializar o processo de contagem dos ciclos de máquina. O botão Synch marca a linha do código em que deve-se iniciar a contagem dos ciclos de máquina e ao executar o programa sincroniza a contagem com a passagem pela linha.

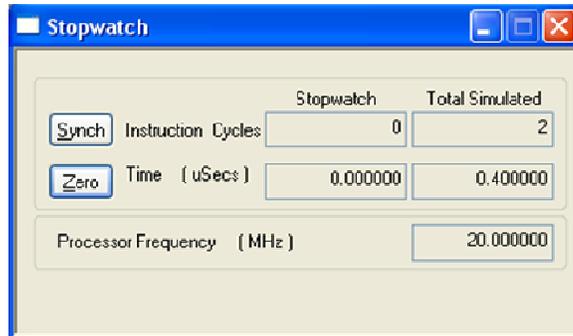


FIGURA 11. JANELA STOPWATCH.

Para fazer a simulação de mudança de estado de um bit, porta, ou comunicação serial utiliza-se a janela Stimulus. Para utiliza-la é necessário criar um arquivo (WorkBook) com as configurações da simulação. Para utilizar Stimulus vá em Debugger → Stimulus → New Workbook. A Figura 12 apresenta a janela Stimulus, sendo utilizado o estímulo assíncrono.

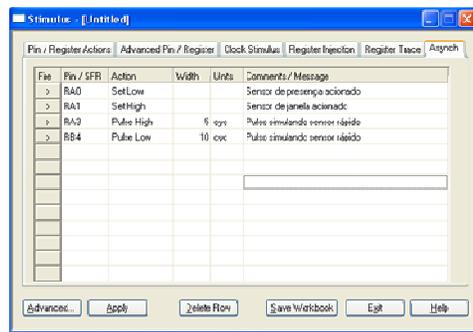


FIGURA 12. JANELA STIMULUS.

Ao criar um novo arquivo de simulação abre-se a janela Stimulus com abas para vários tipos de simulação:

1. Pin/Register Action, simula mudança de estado de bits temporizado e periodicamente.
2. Advanced Pin / Register, simula mudança de estado conforme condições determinadas.
3. Clock Stimulus, simula entrada de um clock no pino.
4. Register Injection, salva o valor selecionado quando o programa está em um ponto determinado.
5. Register Trace, salva o valor selecionado quando certa condição é satisfeita.
6. Assyn, simulação assíncrona, diferente das anteriores, de mudança de estado de bits.

Para utilizar a simulação assíncrona selecione o bit em Pin/SRF, na coluna Action selecione o tipo de ação a ser realizada, sendo elas:

- Set High, leva o estado do bit para 1.
- Set Low, leva o estado do bit para 0.
- Toggle, inverte o estado do bit. Se igual a 1 é feito igual a zero e vice-versa.
- Pulse High, gera um pulso alto (0 → 1 → 0).
- Pulse Low, gera um pulso baixo (1 → 0 → 1).

As colunas Width e Units referem-se à simulação de pulso, onde pode-se selecionar o tempo do pulso a ser simulado. A primeira coluna, Fire, é utilizada para acionar a simulação. Ao ser clicado realiza a respectiva simulação mas só faz efeito após a execução de um ciclo de máquina do programa.

Realizando a Simulação

Após realizar a compilação pode-se configurar a janela Watch para visualizar os registradores mais importantes e configurar mudanças de estado de entradas do microcontrolador através da janela Stimulus. Neste programa pode-se acompanhar as portas de I/O, para visualizar o estado dos LEDs, através da janela Watch e realizar mudança de estado da chave 1, através da janela Stimulus, aba assíncrona. Distribua as janelas na area de trabalho do MPLAB para melhor visualizar e acessar suas funcionalidades, como visto na Figura 13. A configuração da área de trabalho pode ser salva para que possa ser restalrada quando o programa for aberto novamente. Para salvá-la vá em File → Save Workspace.

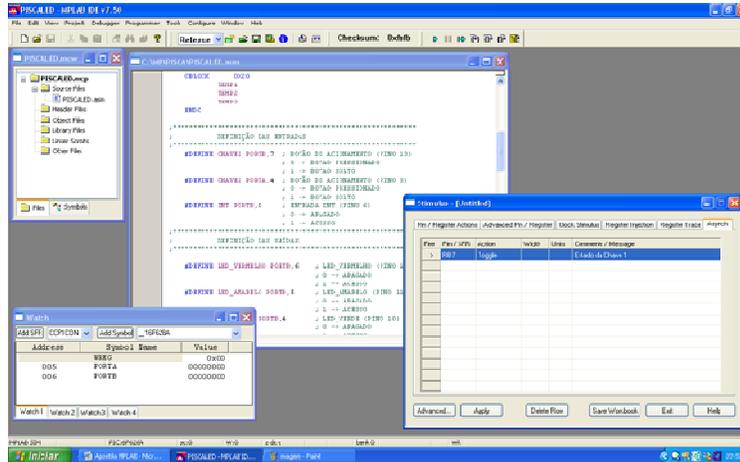


FIGURA 14. DISTRIBUINDO AS JANELAS PARA SIMULAÇÃO.

Um ponto de parada de execução pode ser realizado através de um BreakPoint. Quando o programa estiver em execução e alcançar o ponto com o breakpoint o programa pára e aguarda a continuidade de execução do programa através da intervenção do usuário. Para inserí-lo clique um pouco antes do início da linha onde se quer inserir o breakpoint, aparecerá um círculo vermelho com a letra B dentro, como visto na Figura 14.

do pino RB7, a chave 1. Após a execução de uma instrução, através dos modos Step Into ou Step Over, visualiza-se a mudança de estado do pino RB7, como visto na Figura 16.

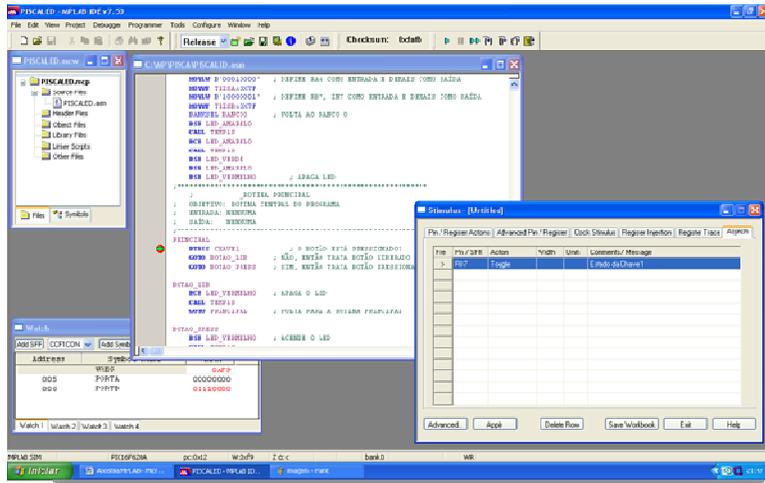


FIGURA 15. PROCESSO DE ANDAMENTO DA SIMULAÇÃO.

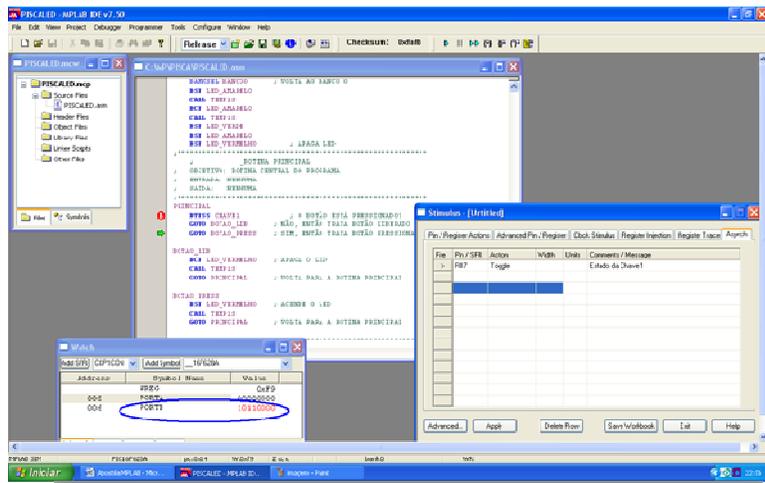


FIGURA 16. ESTÍMULO NA CHAVE 1.

Pode-se observar a diferença entre Step Into e Step Over na chamada da rotina de tempo TEMP15. Quando se utiliza Step Into a simulação entra na rotina e executa-a passo a passo. Quando se utilize Step Over a simulação executa mas não entra na rotina, visualizando-se apenas a barra Stepping... na barra de Status.

Anexo 1. Código do Programa PiscaLed

```
;*****  
;  
;          ROTINA PISCA LED  
;  
;          VERSÃO: 1.0          PIC16F628A          ;  
;          DESCRIÇÃO DO PROGRAMA  
;-----  
;          SISTEMA MUITO SIMPLES PARA REPRESENTAR O ESTADO DE  
;          UM BOTÃO ATRAVÉS DE UM LED  
;*****  
;  
;*****  
;          ARQUIVOS DE DEFINIÇÕES  
;*****  
;          #INCLUDE <P16F628A.INC>          ; ARQUIVO PADRÃO DO  
PIC16F628A          ; PALAVRA DE  
CONFIGURAÇÃO  
;          ;_CONFIG _BOREN_ON & _CP_OFF & _PWRTE_ON & _WDT_OFF & _LVP_OFF & _MCLRE_ON &  
_XT_OSC  
;  
;*****  
;          PAGINAÇÃO DE MEMÓRIA  
;*****  
BANCO0 EQU    0X00    ; ENDEREÇO INICIAL DO BANCO0  
BANCO1 EQU    0X80    ; ENDEREÇO INICIAL DO BANCO1  
BANCO2 EQU    0X100   ; ENDEREÇO INICIAL DO BANCO2  
BANCO3 EQU    0X180   ; ENDEREÇO INICIAL DO BANCO3  
; USAREMOS A DIRETRIZ "BANKSEL" DA LINGUAGEM MPASM PARA SELEÇÃO DO BANCO DE  
REGISTRADORES  
;  
;*****  
;          DEFINIÇÃO DAS VARIÁVEIS  
;*****  
;          ; USANDO A DIRETRIZ "CBLOCK" DEFINIMOS APENAS O PONTO INICIAL DAS VARIÁVEIS  
CBLOCK  
    0X20  
    TEMP1  
    TEMP2  
    TEMP3  
ENDC  
;  
;*****  
;          DEFINIÇÃO DAS ENTRADAS  
;*****  
#DEFINE CHAVE1 PORTB,7; BOTÃO DE ACIONAMENTO (PINO 13)  
;          ; 0 -> BOTAO PRESSIONADO  
;          ; 1 -> BOTAO SOLTO  
#DEFINE CHAVE2 PORTA,4; BOTÃO DE ACIONAMENTO (PINO 3)  
;          ; 0 -> BOTAO PRESSIONADO  
;          ; 1 -> BOTAO SOLTO  
#DEFINE INT PORTB,0    ; ENTRADA INT (PINO 6)  
;          ; 0 -> APAGADO  
;          ; 1 -> ACESSO  
;  
;*****  
;          DEFINIÇÃO DAS SAÍDAS  
;*****  
#DEFINE LED_VERMELHO PORTB,6 ; LED_VERMELHO (PINO 12)  
;          ; 0 -> APAGADO  
;          ; 1 -> ACESSO  
#DEFINE LED_AMARELO PORTB,5    ; LED_AMARELO (PINO 11)  
;          ; 0 -> APAGADO  
;          ; 1 -> ACESSO  
#DEFINE LED_VERDE PORTB,4    ; LED_VERDE (PINO 10)  
;          ; 0 -> APAGADO  
;          ; 1 -> ACESSO  
#DEFINE PWM    PORTB,3    ; SAIDA PWM (PINO 9)
```

; 0 -> APAGADO
; 1 -> ACESSO

```
*****  
; VETOR DE RESET  
*****  
ORG 0X00 ; VAI PARA O ENDEREÇO INICIAL DO PROGRAMA  
GOTO INICIO  
  
*****  
; _INICIO  
; OBJETIVO: INICIALIZAÇÃO DO SISTEMA  
; ENTRADA: NENHUMA  
; SAÍDA: NENHUMA  
*****  
INICIO  
CLRF PORTA ; LIMPA O PORTA  
CLRF PORTB ; LIMPA O PORTB  
BANKSEL BANCO1 ; ALTERA PARA O BANCO1  
MOVLW B'00010000' ; DEFINE RA4 COMO ENTRADA E DEMAIS COMO SAÍDA  
MOVWF TRISA&0X7F  
MOVLW B'10000001' ; DEFINE RB7, INT COMO ENTRADA E DEMAIS COMO SAÍDA  
MOVWF TRISB&0X7F  
BANKSEL BANCO0 ; VOLTA AO BANCO 0  
BSF LED_AMARELO  
CALL TEMP1S  
BCF LED_AMARELO  
CALL TEMP1S  
BSF LED_VERDE  
BSF LED_AMARELO  
BSF LED_VERMELHO ; APAGA LED  
*****  
; _ROTINA PRINCIPAL  
; OBJETIVO: ROTINA CENTRAL DO PROGRAMA  
; ENTRADA: NENHUMA  
; SAÍDA: NENHUMA  
*****  
PRINCIPAL  
BTSS CHAVE1 ; O BOTÃO ESTÁ PRESSIONADO?  
GOTO BOTAO_LIB ; NÃO, ENTÃO TRATA BOTÃO LIBERADO  
GOTO BOTAO_PRESS ; SIM, ENTÃO TRATA BOTÃO PRESSIONADO  
  
BOTAQ_LIB  
BCF LED_VERMELHO ; APAGA O LED  
CALL TEMP1S  
GOTO PRINCIPAL ; VOLTA PARA A ROTINA PRINCIPAL  
  
BOTAQ_PRESS  
BSF LED_VERMELHO ; ACENDE O LED  
CALL TEMP1S  
GOTO PRINCIPAL ; VOLTA PARA A ROTINA PRINCIPAL  
  
*****  
; ROTINAS DE TEMPO  
*****  
  
TEMP1S  
MOVLW .6  
MOVWF TEMP1  
TEMP1S_LOOP1  
MOVLW .222  
MOVWF TEMP2  
TEMP1S_LOOP2  
MOVLW .249  
MOVWF TEMP3  
TEMP1S_LOOP3  
DECFSZ TEMP3  
GOTO TEMP1S_LOOP3  
DECFSZ TEMP2  
GOTO TEMP1S_LOOP2  
DECFSZ TEMP1  
GOTO TEMP1S_LOOP1  
RETURN  
*****  
; _FIM DO PROGRAMA  
*****  
END
```