

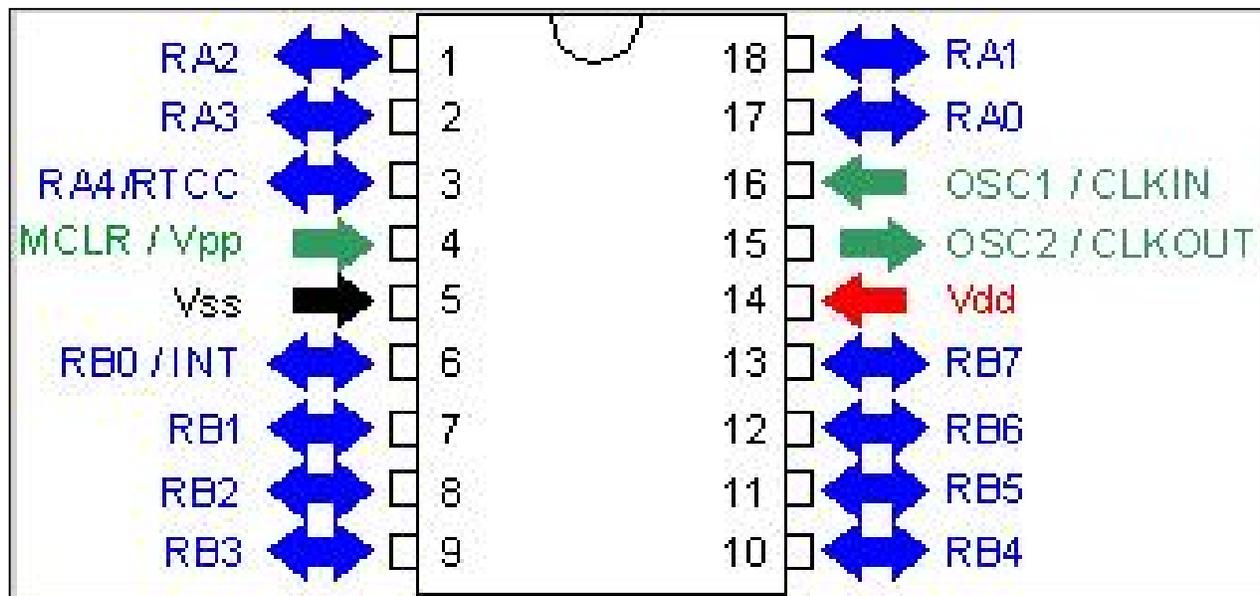
MICROCONTROLADORES

Os microcontroladores são chips inteligentes, que tem um processador, pinos de entradas/saídas e memória. Através da programação dos microcontroladores podemos controlar suas saídas, tendo como referencia as entradas ou um programa interno.

O que diferencia os diversos tipos de microcontroladores, são as quantidades de memória interna (programa e dados), velocidade de processamento, quantidade de pinos de entrada/saída (I/O), alimentação, periféricos, arquitetura e set de instruções.

O QUE É O PIC?

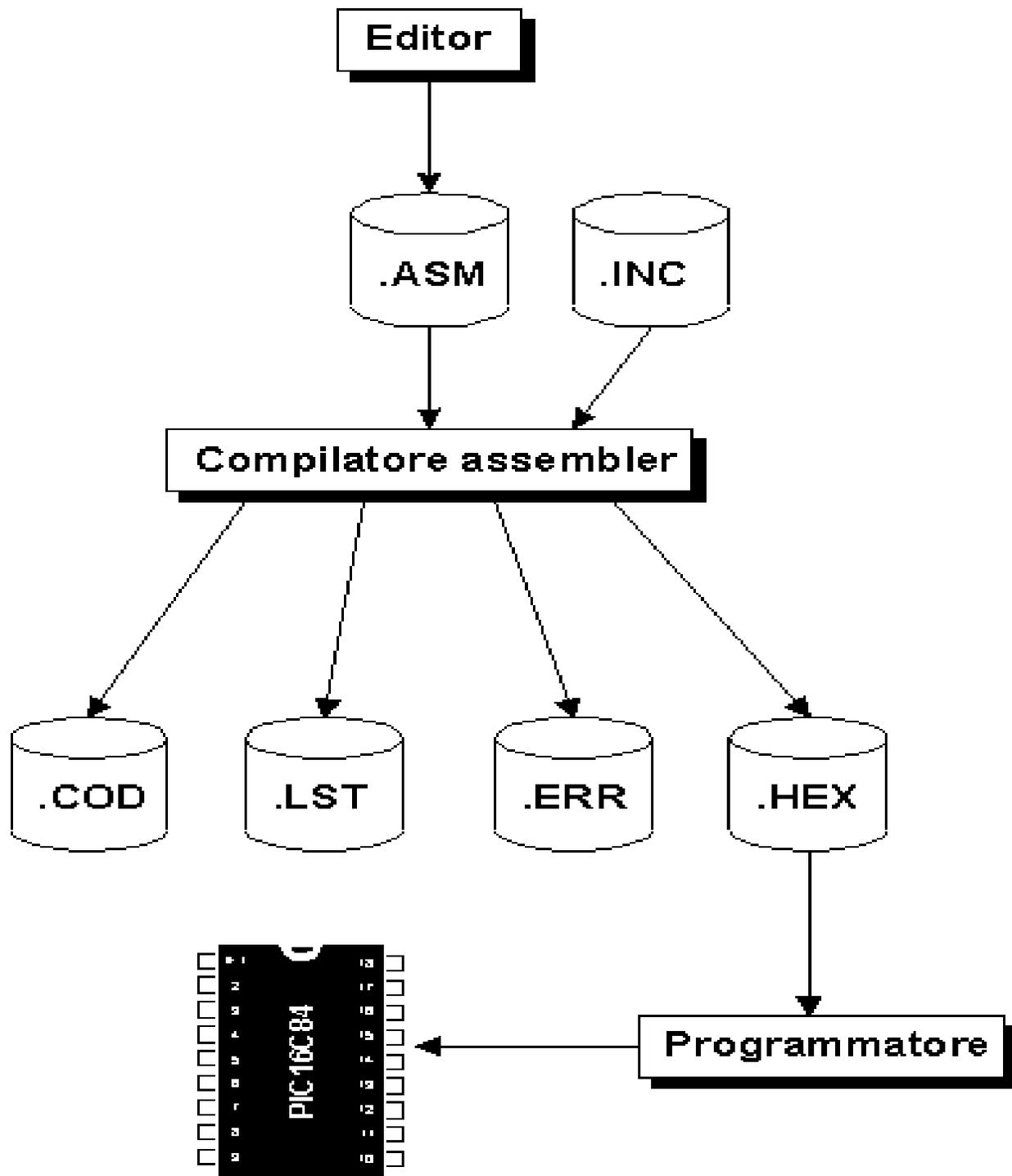
O PIC é um circuito integrado produzido pela Microchip Technology Inc. , que pertence a categoria dos microcontroladores, ou seja, um componente integrado que em um único dispositivo contem todos os circuitos necessarios para realizar um completo sistema digital programavel. O pic pode ser visto externamente como um circuito integrado TTL ou CMOS normal, mas internamente dispõe de todos os dispositivos típicos de um sistema microprocessado, ou seja: Uma CPU (Central Processor Unit ou Unidade de Processamento Central) e sua finalidade é interpretar as instruções de programa; Uma memória PROM (Programmable Read Only Memory ou Memória Programavel Somente para Leitura) na qual ira memorizar de maneira permanente as instruções do programa; Uma memória RAM (Random Access Memory ou Memória de Acesso Aleatório) utilizada para memorizar as variaveis utilizadas pelo programa; Uma serie de LINHAS de I/O (entrada e saída) para controlar dispositivos externos ou receber pulsos de sensores, chaves, etc. ; Uma serie de dispositivos auxiliares ao funcionamento, ou seja, gerador de clock, bus, contador, etc. A presença de todos estes dispositivos em um espaço extremamente pequeno, da ao projetista ampla gama de trabalho e enorme vantagem em usar um sistema microprocessado, onde em pouco tempo e com poucos componentes externos podemos fazer o que seria oneroso fazer com circuitos tradicionais.O PIC esta disponível em uma ampla gama de modelos para melhor adaptar-se as exigências de projetos específicos, diferenciando-se pelo numero de linha de I/O e pelo conteúdo do dispositivo. Inicia-se com modelo pequeno identificado pela sigla PIC12Cxx dotado de 8 pinos, até chegar a modelos maiores com sigla PIC17Cxx dotados de 40 pinos. Uma descrição detalhada da tipologia do PIC é disponível no site da Microchip (www.microchip.com) , onde conseguimos encontrar grandes e variadas quantidades de informações técnicas, software de apoio, exemplos de aplicações e atualizações disponíveis.



O PROCESSO DE GRAVAÇÃO DE UM PIC

Para podermos escrever (gravar) em um microcontrolador nós vamos precisar de uma gravadora, um software gravador, um compilador e um programa.

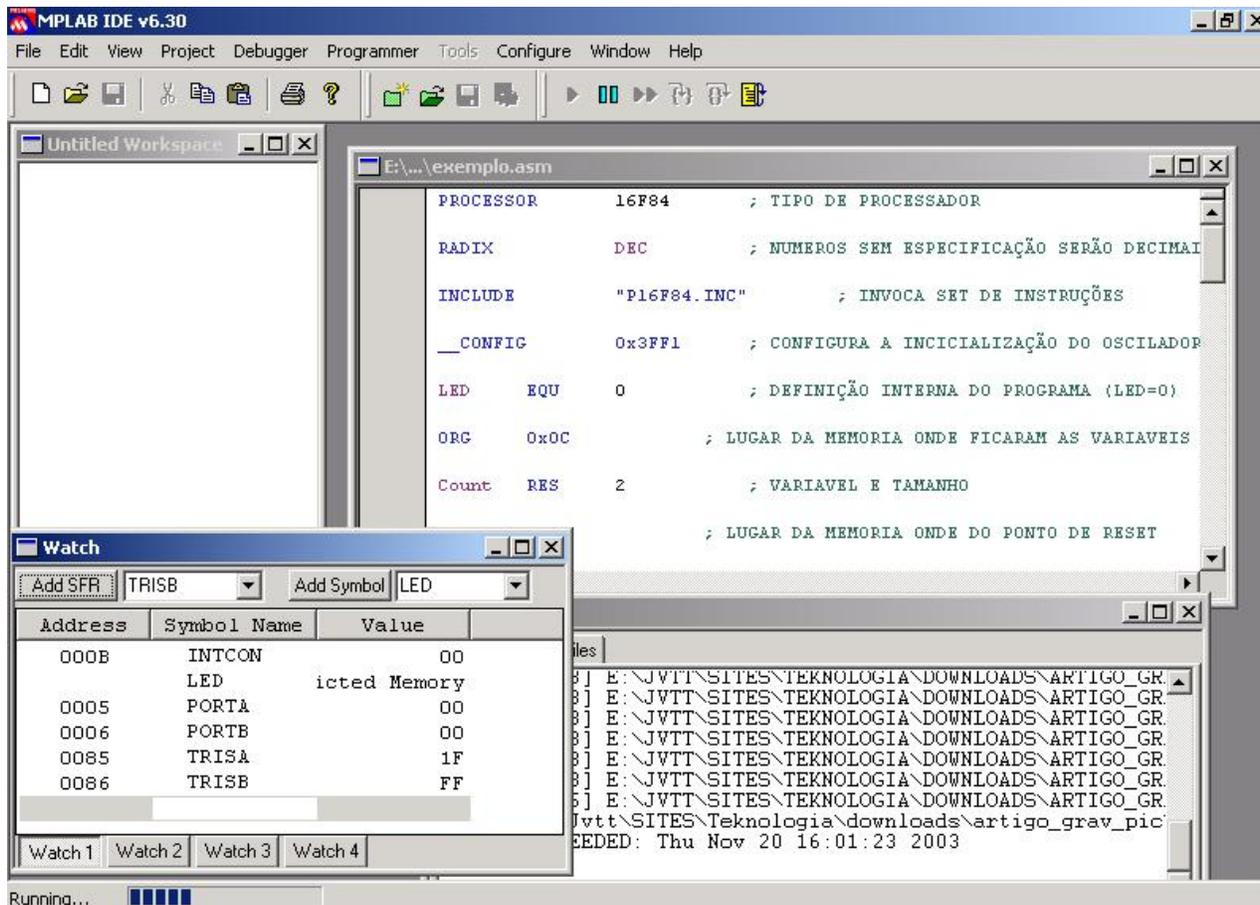
- O programa pode ser escrito em assembler. Nele ira conter as informações de configuração do nosso microcontrolador e a lógica do nosso dispositivo. Normalmente usa-se o bloco de notas ou o MPLAB IDE (editor e simulador) para escrever estes programas.
- O compilador (mpasmwin – que pode ser encontrado no site <http://www.microchip.com>) vai transformar as informações geradas pelo programa (*.asm) em opcode – códigos operacionais (hexadecimais de 14 bits) e transformar em hexadecimal (*.hex).
- O Software Gravador (icprog – www.ic-prog.com), vai transmitir as informações hexadecimais geradas pelo compilador de forma correta para o gravador.
- O Gravador converte e organiza os sinais gerados pelo computador, para que eles sejam armazenados no microcontrolador.



Também podemos simular nosso projeto através do software MPLAB IDE, onde podemos monitorar todos os endereços de memória e possibilita o acompanhamento de todo o sistema, para identificar e solucionar problemas, o

software pode ser encontrado no endereço abaixo.

<http://www.microchip.com/1010/pline/tools/picmicro/devenv/mplabi/mplab6/index.htm>



ESCREVENDO UM PROGRAMA

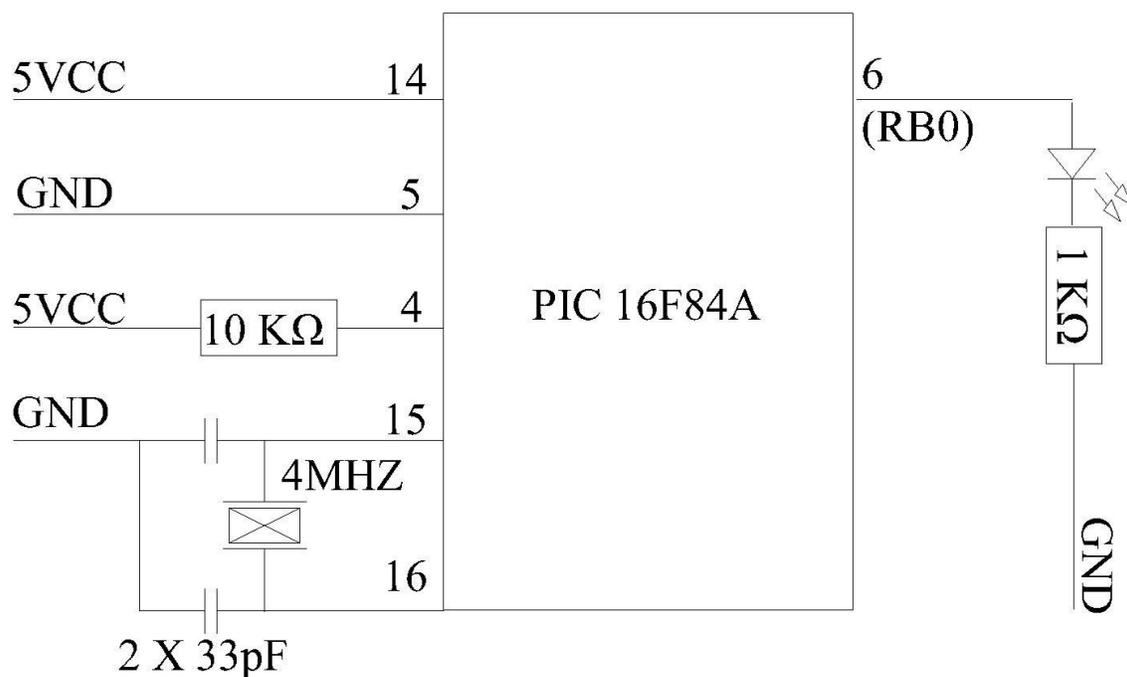
Fazem parte de um programa para nosso microcontrolador: o tipo do processador (ex. p16f84a), arquivo contendo o set de instruções – comandos (ex. p16f84.inc), o tipo de oscilador e recursos de gravação, as portas que serão de entrada/saída, as variáveis do sistema, os endereços de memória que cada parte do programa vai utilizar, e o mais importante: A lógica de programação.

Para aprender a programar no pic, não deixe de visitar esta excelente pagina: <http://tanzilli.com/pbe/brasiliano/index.htm> (em português) ou baixe o curso em <http://www.mecatrom.cjb.net/> , na seção de downloads.

Segue abaixo um exemplo comentado de programa:

```
PROCESSOR    16F84           ; TIPO DE PROCESSADOR
RADIX        DEC            ; NUMEROS SEM ESPECIFICAÇÃO SERÃO
                                DECIMAIS
INCLUDE      "P16F84.INC"   ; INVOCA SET DE INSTRUÇÕES
__CONFIG     0x3FF1         ; CONFIGURA A INCIALIZAÇÃO DO OSCILADOR
LED EQU     0               ; DEFINIÇÃO INTERNA DO PROGRAMA (LED=0)
ORG          0x0C           ; LUGAR DA MEMORIA ONDE FICARAM AS
                                VARIAVEIS

Count RES    2              ; VARIAVEL E TAMANHO
ORG          0x00           ; LUGAR DA MEMORIA ONDE DO PONTO DE RESET
bsf STATUS,RP0             ; ACESSA O BANCO "0" DE NDEREÇOS/COMANDOS
movlw B'00011111'         ; ADICIONA ESTE VALOR PARA WORKS (W)
movwf TRISA                ; DEFINE AS ENTRADAS/SAIDAS DO PORTA
movlw B'11111110'         ; ADICIONA ESTE VALOR PARA WORKS (W)
movwf TRISB                ; DEFINE AS ENTRADAS/SAIDAS DO PORTB
bcf STATUS,RP0            ; VOLTA P/ BANCO "1" DE ENDEREÇOS/COMANDOS
bsf PORTB,LED              ; DESLIGA O PINO 6 (RB0) DO CI, APAGANDO O LED
MainLoop                  ; DECLARAÇÃO DE ROTINA PRINCIPAL
call Delay                 ; CHAMADA DE ROTINA DE TEMPO
btfsc PORTB,LED            ; TESTA SE O LED ESTA ACESO (1 OU 0)
goto SetToZero             ; SE APAGADO ENTÃO EXECUTA ESTA ROTINA
bsf PORTB,LED              ; SE NÃO, APAGA O LED (RBO = 0)
goto MainLoop              ; VOLTA PARA A ROTINA PRINCIPAL
SetToZero                  ; ROTINA QUE ACENDE O LED
bcf PORTB,LED              ; COMANDO QUE ACENDE O LED
goto MainLoop              ; VOLTA PARA ROTINA PRINCIPAL
Delay                       ; ROTINA DE TEMPO
clrf Count                 ; LIMPA COUNT
clrf Count+1               ; LIMPA COUNT+1
DelayLoop                  ; ROTINA DE LOOP DE TEMPO
decfsz Count,1             ; DECREMENTA E PULA A PROXIMA LINHA SE 0
goto DelayLoop             ; VOLTA PARA O LOOP DE TEMPO
decfsz Count+1,1           ; DECREMENTA E PULA A PROXIMA LINHA SE 0
goto DelayLoop             ; VOLTA PARA O LOOP DE TEMPO
return                      ; VOLTA PARA ONDE CHAMOU A ROTINA
END                          ; FINALIZA O PROGRAMA
```



Para as demais experiências com microcontroladores pic, recomendo a compra dos seguintes materiais :

- 2 capacitor 33pF (necessario para a geração de clock)
- 1 cristal 4 MHz (necessario para a geração de clock)
- 6 resistor 10K (5 de entrada para o PORTA + 1 para o RESET)
- 8 resistor 1K (para o PORTB)
- 6 Chave touch 4mm (5 de entrada para o PORTA + 1 para o RESET)
- 8 led vermelho (para poder visualizar o estado do PORTB)
- 1 soquete dip torneado 18 pinos (p/ deixar fixo no CI do PIC, evitando quebra)
- 1 microcontrolador PIC16F84A (ótimo para começar !!!!)

O COMPILADOR – MPASMWIN

O compilador vai transformar o código fonte do programa em novas instruções sequenciais para o microcontrolador. O mpasmwin encontra-se em versões para windowsn (mpasmwin) e para ms-dos (mpasmdos), pode se baixado em <http://www.microchip.com/1010/pline/tools/archive/other/61/index.htm> , www.microchip.com ou junto com o MPLAB IDE .

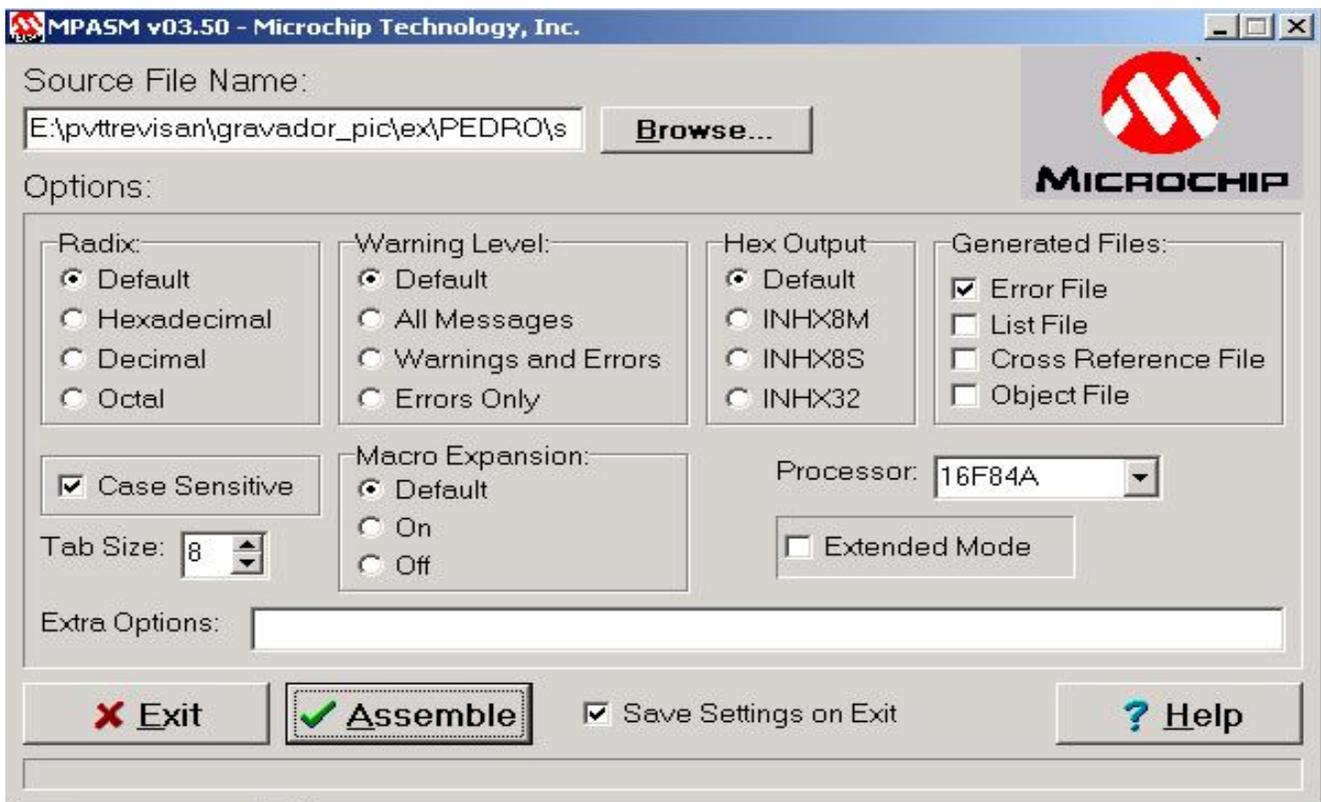
Cada uma das instruções identificara precisamente a função básica que o PIC irá executar. Onde a instrução é representada por um código operativo (do inglês operation code ou abreviadamente opcode) podemos memorizar 14 bits em cada localização da memória EEPROM. Esta memória no PIC16C84 dispõe de 1024 localizações e cada uma devere conter uma só instrução. Um exemplo de opcode em notação binária está escrito a seguir: 00 0001 0000 0000B

É mais provável que um opcode venha representado na notação hexadecimal ou seja: 0100H

Este código, completamente sem sentido para nós humanos, é o que o PIC está preparada para entender. Para facilitar a compreensão ao programador, se recorre a um instrumento e convenção para tornar a instrução mais compreensível.

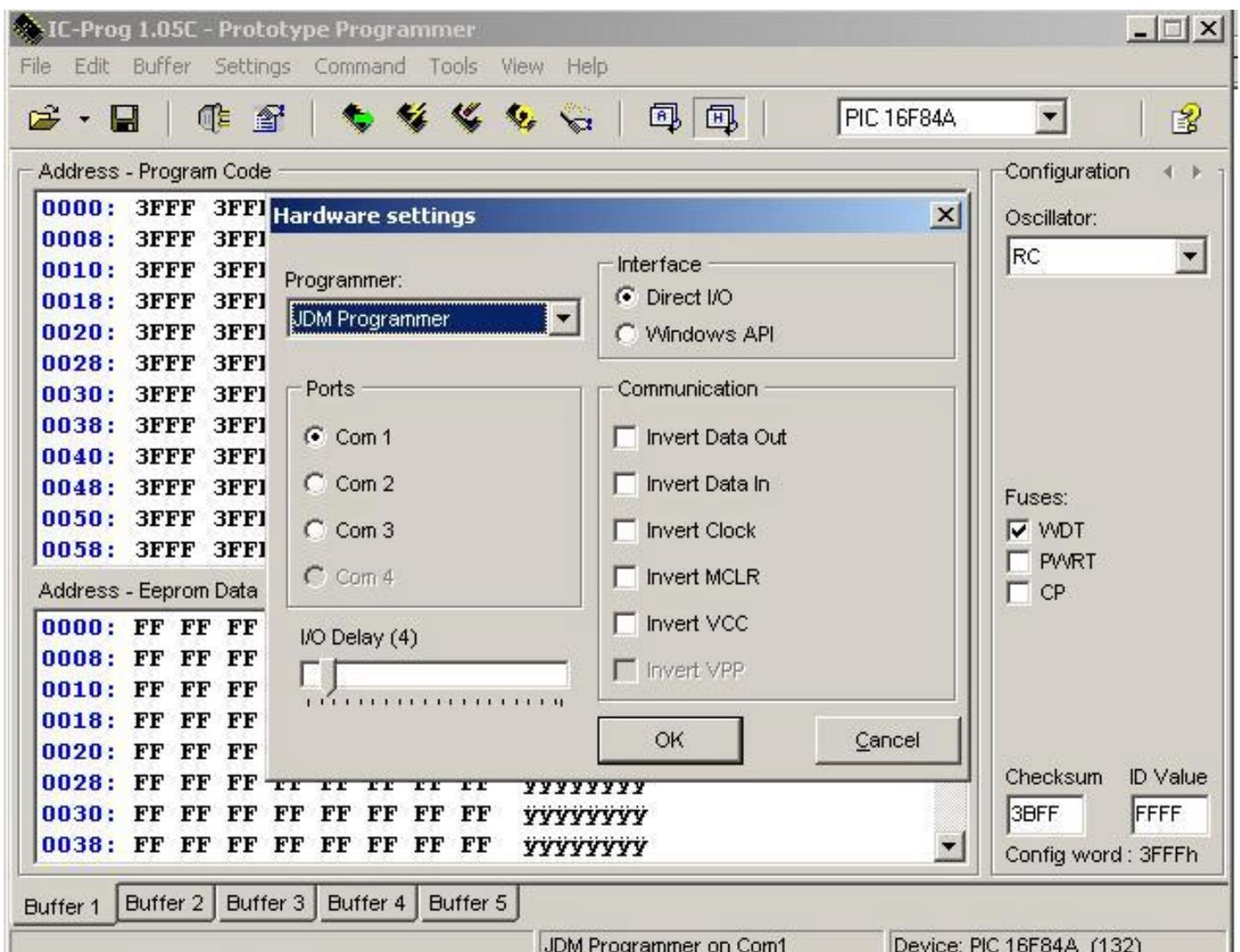
A primeira convenção é a que associa o opcode (um total de 35 para o PIC16C84) a uma sigla mnemônica, ou seja uma inicial que seja fácil de recordar o significado da instrução.

Voltando ao nosso exemplo o opcode 0100H corresponde a instrução mnemônica CLRW que é a forma abreviada da instrução CLEAR W REGISTER, ou seja, zere o registro W. Estes códigos podem ser encontrados dentro do arquivo de biblioteca do compilador, que é invocado pelo programa. No caso este arquivo é o P16C84 (pela linha include "P16C84.INC).



O SOFTWARE GRAVADOR - ICPROG

Através do ic-prog (<http://www.ic-prog.com/icprog105C.zip> ou www.ic-prog.com) vamos transferir o arquivo gerado pelo compilador (que agora é hexadecimal - *.hex) para a nossa gravadora. Existem varios tipos de gravadoras compatíveis com o icprog. No nosso caso vamos utilizar o tipo JDM. Para isso va ao menu configurações e na opção Hardware (ou pressione F3) e configure o programa como a figura abaixo (atenção, para os sistema 2000/NT/XP, leia o texto em negrito após a figura) :



No menu configurações, opções temos a possibilidade de substituir o idioma para português (guia linguagem) e de ativarmos o suporte para

windows NT/2000/XP. Para isso você vai precisar baixar também o arquivo “icprog.sys” (http://www.ic-prog.com/icprog_driver.zip ou www.ic-prog.com) para a pasta de instalação do icprog e em seguida salvar e sair. Reinicie o programa e vá para o menu configuração e em seguida hardware (F3) e confirme a tela a cima.

O GRAVADOR DE PIC'S

O gravador é o hardware, que vai converter os impulsos eletrônicos ativados pelo software gravador (ic-prog) na porta seriais/paralela para os pinos do nosso microcontrolador. Esta gravação é feita de forma serial, ou seja após codificar os comandos em números hexadecimais, agora transcodificamos-os em impulsos elétricos seriais para a gravadora, que vai atingir os níveis de tensões necessários para possibilitar a transferência dos códigos para dentro da memória de dados do PIC.

O princípio de gravação é o mesmo para todas as gravadoras, o que muda é o hardware que o desenvolvedor vai utilizar para garantir segurança para seu computador, seu microcontrolador e sua gravadora.

Em algumas gravadoras temos a fonte na própria placa e em outras utilizaremos a tensão existente na porta de comunicação do micro (em geral, uns 15mA). Normalmente em uma fonte para a gravadora temos a ponte retificadora, um capacitor de 1000uF-eletrolítico, um de 100nF-eletrolítico, em seguida um LM7805 (TO-220) e um capacitor de 100uF-eletrolítico. Portanto, durante o processo de gravação o pic deve ser alimentado com +5Vcc e ter o GND conectado.

Em determinado momento da gravação precisaremos de um pulso entre 13Vcc e 15Vcc no pino 5 - MCLR (Memory Clear), portanto também teremos uma outra fonte para gerar esta tensão.

O pino SDA (RB7) transmite e recebe dados. É por este pino que os dados a serem gravados são inseridos ou lidos, conforme o comando selecionado.

O pino SCL (RB6) é o clock de sincronismo das informações.

O grande segredo é saber onde enviar estes pulsos (veja a tabela abaixo).

PINO	DAS	SCL	MCLR	VCC	GND
SOQUETE 8	7	6	4	1	8
SOQUETE 18	13	12	4	14	5
SOQUETE 28	28	27	1	20	8
SOQUETE 40	40	39	1	11/32	12/31

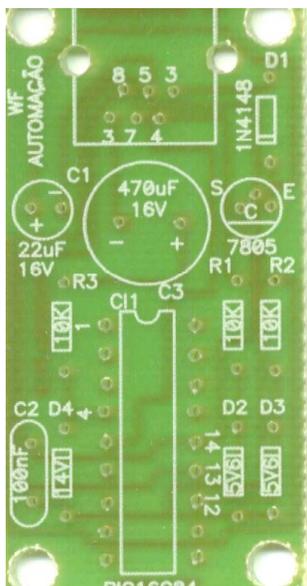
Atenção, apesar dos PIC's serem praticamente imunes a energia estatica e muito resistentes para qualquer ambiente, nunca devemos inserir ou retirar o microcontrolador com a gravadora ligada ou conectada. Para preservação da porta de comunicação com o pc, não devemos remover o cabo durante o processo de gravação ou com o ic-prog aberto. É recomendavel que você mantenha sempre o microcontrolador em um porta soquete (mesa para CI) com furos torneados, aumentando assim a vida útil do microcontrolador.

EXEMPLO 1

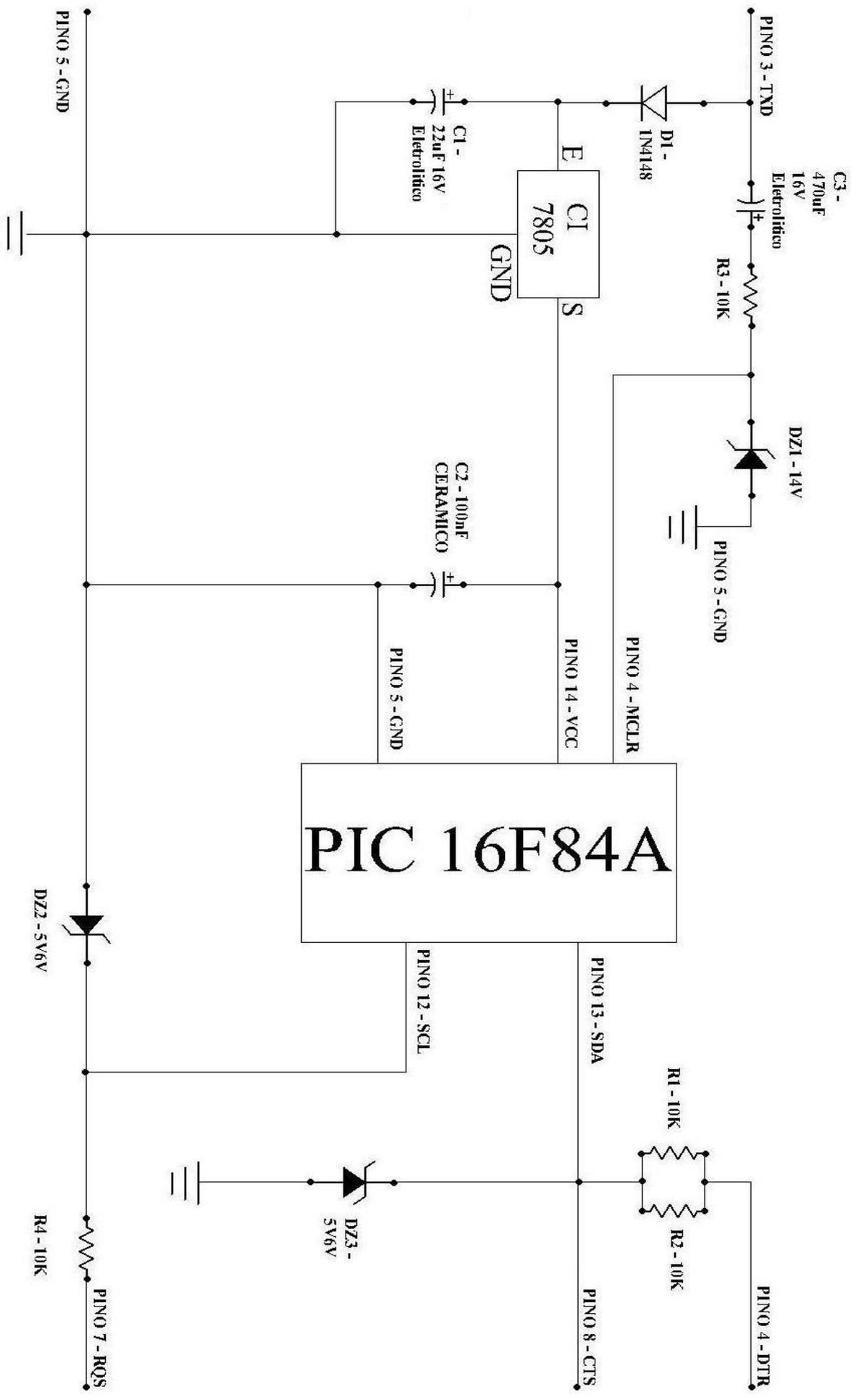
Gravador exclusivo para pic's de 18 pinos (PIC16C84, PIC16f84, PIC16F84A, PIC16F628, etc). Visite a ótima pagina <http://www.furb.br/~mw/pratica/>, para mais informações. Este é o gravador que eu uso diariamente e recomendo-o.

Ele obtém os 13VCC para o MCLR de um diodo zener e os 5VCC de alimentação de um 7805. Os outros zeners (5,6V) são para controlar a tenção nos pinos de SDA e SDL. Este modelo, não precisa de fonte externa.

O software a ser utilizado é o IC-PROG (<http://www.ic-prog.com/>), com as configurações ja mencionadas a cima.



- 2 zeners 5v6
- 4 res 10k
- 1 7805 (formato transistor - BC)
- 1 1n4148
- 1 cap eletrolitico 22uF/16v
- 1 cap eletrolítico 470uF/16v
- 1 zener 13V
- 1 cap 100nF cerâmico
- 1 soquete dip18 torneado
- 1 conector DB9 fêmea + capa
- 1,5 mts de cabo 5 vias com malha
- 1 kit parafusos espaçadores (base do gravador)
- 1 PIC 16f84A - 4p



EXEMPLO 2

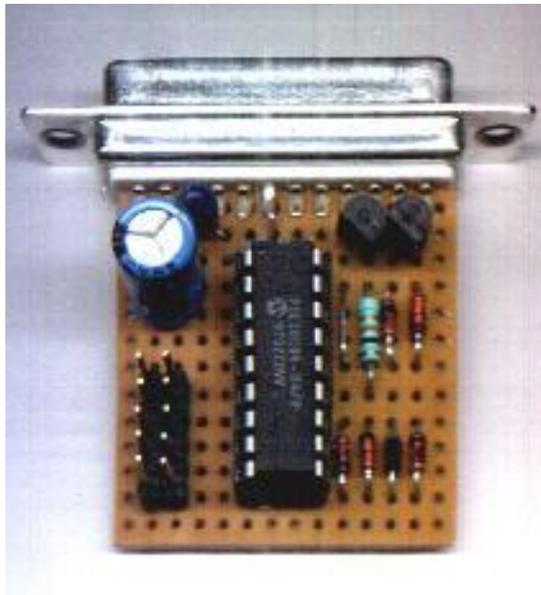
Este modelo é um dos mais usados nos EUA, é conhecido como JDM. Podemos dizer que muitos dos gravadores existentes foram baseados neste modelo. Na página oficial, você vai encontrar a layout para placa de circuito impresso e muitas outras informações (em inglês, é claro).

<http://www.jdm.homepage.dk/>

<http://www.jdm.homepage.dk/newpic.htm>

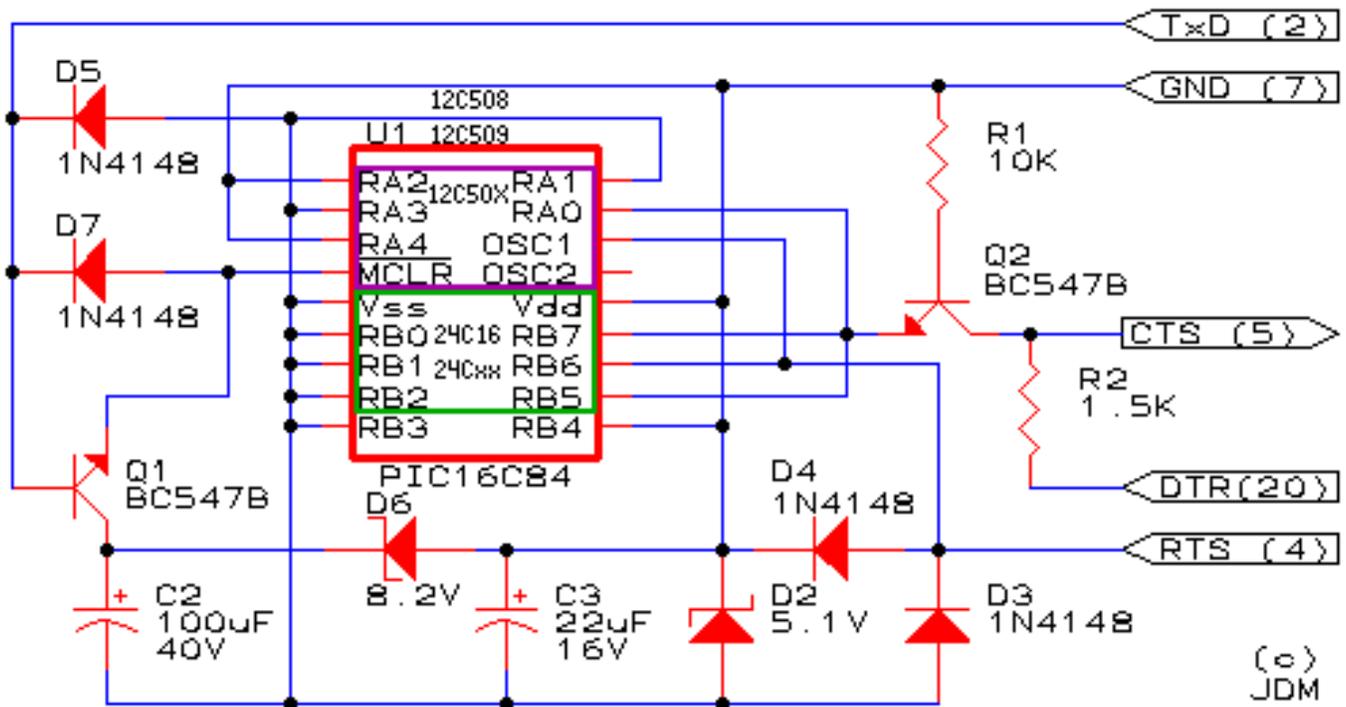
Este gravador é alimentado por uma entrada RS-232 e trabalha com níveis de tensões próprios desta configuração. O nível de 14VCC é conseguido através de um zener de 8,2V e um ganho de 3V com o transistor. Ele grava PIC12C5XX, 12C67X, 24CXX, 16C55X, 16C61, 16C62X, 16C71, 16C71X, 16C8X, 16F8X. A voltagem do circuito é estabilizada com diodos zener. Este modelo não precisa de fonte externa e pode ser utilizado para gravação in circuit (sem remover o pic do circuito). Observe que ele trabalha com a antiga porta serial de 25 pinos, para utiliza-la em micros novos com o conector DB9 Femea (igual do mouse) utilize: TXD (3), GND (5), DTR (4), CTS (8), RQS (7).

O software a ser utilizado pode ser o IC-PROG (<http://www.ic-prog.com/>), ou um outro indicado pelo desenvolvedor , consulte o site oficial do gravador e do ic-prog, para maiores detalhes.



R1 10K	Resistor
R2 1.5K	Resistor
D2 5.1V/0.5W	Zener
D3 1N4148 ou 1N4448	Diodo
D4 1N4148 ou 1N4448	Diodo
D5 1N4148 ou 1N4448	Diodo
D6 8.2V/0.5W	Zener
D7 1N4148 ou 1N4448	Diodo
C2 100µF/25V	Capacitor eletrolitico
C3 22µF/16V ou 47µF/6.3V	Capacitor tantaló
Q1 BC547B	Transistor NPN
Q2 BC547B	Transistor NPN
P1 DS25 (female)	25 pol DSUB connector
P2 Conector	Conector para programação no circuito

Não esqueça dos 2 jumpers em baixo da placa



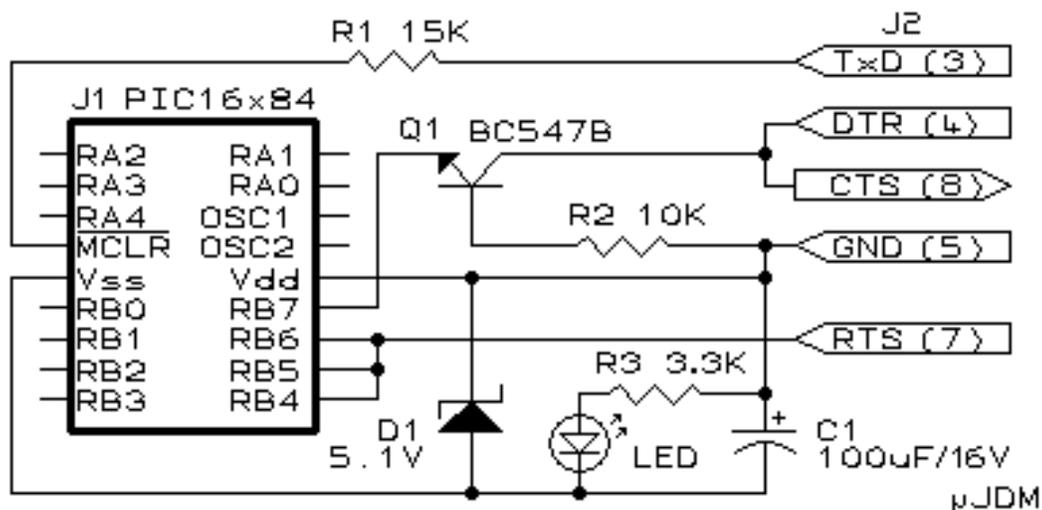
EXEMPLO 3

Um outro bom exemplo de gravador pode ser obtido em:

<http://www.semis.demon.co.uk/>

<http://www.semis.demon.co.uk/uJDM/uJDMmain.htm>

Ele foi desenvolvido com base no JDM original, porém tem seu funcionamento simplificado. É conhecido como micro JDM (uJDM).



TODOS OS PROGRAMAS, AQUI MENSIONADOS SÃO “FREEWARE”, OU SEJA, POSSUEM LIVRE DISTRIBUIÇÃO. NÃO ME RESPONSABILIZO POR QUALQUER DANO CAUSADO EM SEU EQUIPAMENTO.

NOSSA INTENÇÃO NÃO FOI A DE APROFUNDAR NAS TÉCNICAS DE PROGRAMAÇÃO DE MICROCONTROLADORES, MAS SIM CATIVAR O INTERESSE DO LEITOR PELO FACINANTE MUNDO DA AUTOMAÇÃO, TECNOLOGIA E MICROCONTROLADORES.

ABRAÇOS,
PEDRO V. T. TREVISAN
(pvttrevisan@ig.com.br)

VISITE MEU SITE
<http://www.mecatrom.cjb.net/>