

Programação em BASIC para o PIC

Vitor Amadeu Souza (vitor@cerne-tec.com.br)

Introdução

Os microcontroladores da família PIC estão a cada dia mais presentes nos projetos eletrônicos. Em média, cada um de nós têm acesso a 9 chips desses todos os dias e aprender a trabalhar e programar estes *computadores em um só chip* se torna praticamente imperativo para o profissional do campo tecnológico. Aplicações como robótica, domótica, automação, entretenimento e etc são alguns dos campos de utilização destes chips.

Existem diversas linguagens no qual o PIC pode ser programado. Dentro deste leque, irei abordar nesta série de artigos a linguagem BASIC com base no compilador mikroBASIC desenvolvido pela mikroElektronika. Este artigo será baseado no livro recém escrito de minha própria autoria, chamado *Programação em BASIC para o microcontrolador PIC18F1220* (Editora Érica 2006) que pode ser verificado na figura 1.



Figura 1 – Livro Programação em BASIC para o microcontrolador PIC18F1220

Este livro pode ser adquirido nas melhores livrarias técnicas ou através do site da editora (www.ERICA.com.br). Recomendo a todos que querem ter um conhecimento mais aprofundado da linguagem e do PIC que façam a leitura deste livro. Informo aos leitores desta revista que acompanharão esta série que baixem o compilador no site do fabricante, no endereço www.mikroe.com. A versão disponível para download é chamada de versão DEMO. Nesta versão, o código máximo que o compilador irá gerar é de 2kW e em todos os projetos desta revista, em nenhum momento este limite será ultrapassado.

O kit de desenvolvimento utilizado para testes é placa PICLAB18F1220 que utiliza o microcontrolador PIC18F1220 e pode ser verificado na Figura 2. Este kit foi desenvolvida pela Cerne Tecnologia e Treinamento (www.cerne-tec.com.br), empresa no qual eu represento.

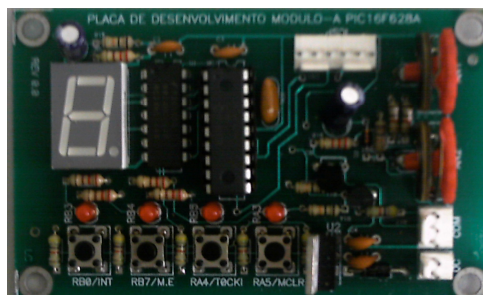


Figura 2 – Placa de Desenvolvimento PICLAB18F1220

O PIC

Os microcontroladores da família PIC18 apresentam-se com grandes vantagens frente a família PIC16. Aspectos como memória de programa e dados são apenas alguns dos vários itens que fazem da família PIC18 superior a PIC16. Observe a tabela 1, onde é apresentado um quadro comparativo entre estas duas famílias.

Descrição	PIC16	PIC18
Memória de programa	8kW *	2 MB *
Memória de dados	512 B *	4 kB *
Interrupção	Único vetor sem prioridades	Dois vetores sendo um de alta prioridade e outro de baixa prioridade
USB	Versão 1.1	Versão 2.0
Módulo LVD (Low Voltage Detect)	Não	Sim
Memória de programa linear	Não, é Segmentada	Sim
Timers	3 *	4 *
Processamento	5 MIPS *	10 MIPS *

* Valores máximos

Tabela 1 – Comparativo entre a família PIC16 e PIC18

Note neste pequeno quadro que a família PIC18 é bem superior a família PIC16. Outro detalhe importante é que mesmo sendo melhor, o custo de ambas as famílias está muito próxima.

O microcontrolador utilizado neste artigo para demonstrar os aspectos da família PIC será o PIC18F1220 onde a pinagem deste componente pode ser observada na figura 2.

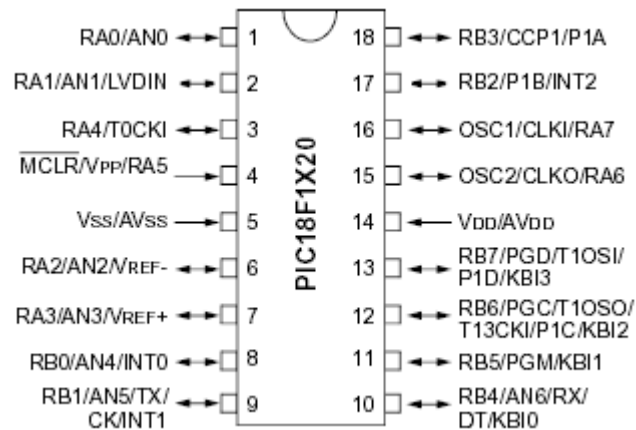


Figura 2 – Microcontrolador PIC18F1220

Observe que como nas famílias tradicionais de 18 pinos, este microcontrolador possui dois PORTS denominados PORTA e PORTB. O PORTA está ligado aos pinos 1 (RA0), 2 (RA1), 3 (RA4), 4 (RA5), 6 (RA2), 7 (RA3), 15 (RA6) e 16 (RA7). Já os pinos do PORTB estão ligados aos pinos 8 (RB0), 9 (RB1), 17 (RB2), 18 (RB3), 10 (RB4), 11 (RB5), 12 (RB6) e 13 (RB7). Este microcontrolador funciona com uma tensão de 2 V até 5,5 V. Em todos os exemplos a alimentação será de 5 V.

Vejamos algumas características importantes deste microcontrolador na tabela 2.

Característica
Memória de Programa Flash de 2 kW
Memória de dados RAM de 256 bytes
Memória de dados EEPROM de 256 bytes
16 pinos de I/O
Fonte de clock interna de até 8 MHz
Módulo LVD
15 Fontes de interrupção
USART
Módulo CCP
Conversor analógico para digital

Tabela 2 – Características do PIC18F1220

Este microcontrolador está instalado na placa didática PICLAB18F1220 e nesta placa poderemos testar praticamente todos os seus recursos.

O mikroBASIC

Para que possamos criar qualquer programa neste ambiente, é necessário primeiramente criar um projeto. Para isso, após o download e instalação do mikroBASIC, inicialize o mesmo e vá no menu *Project -> New Project*. A tela apresentada na figura 3 surgirá.

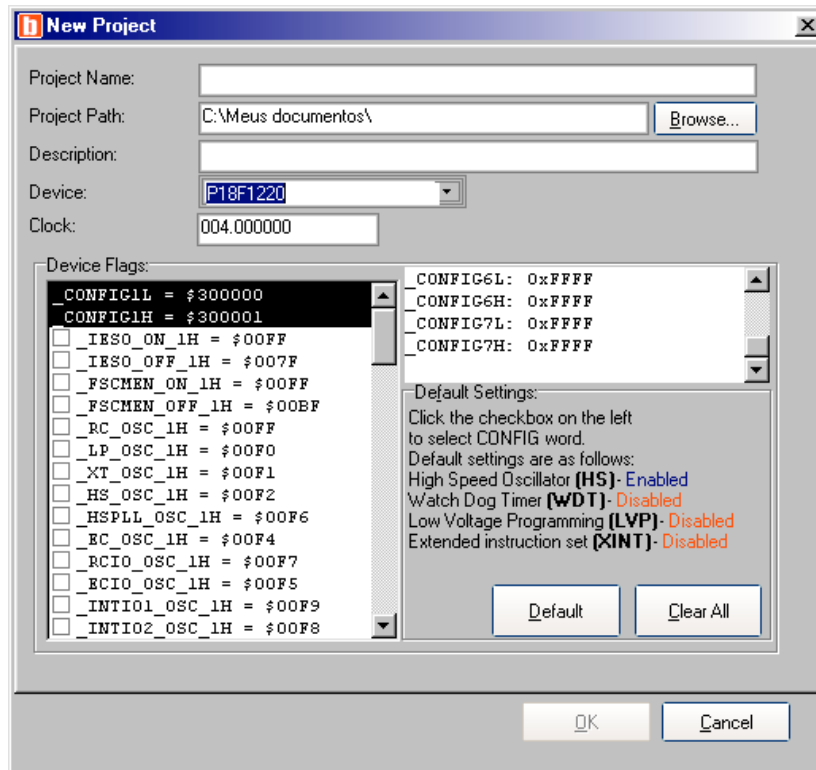


Figura 3 – Criando um novo projeto no mikroBASIC

No campo *Project Name* é informado o nome do seu projeto. Neste campo, digite por exemplo *experimento1*. No campo *Project Path* definimos a pasta onde os arquivos referentes ao projeto ficarão salvos. Escolha neste caso, a pasta de sua preferência. O campo *description* é opcional e serve para fazermos alguma descrição referente ao projeto, sobre o que o mesmo faz por exemplo. Em *device* é escolhido o microcontrolador utilizado no projeto. Clique neste campo e note quantos microcontroladores o mikroBASIC suporta em sua lista. Como estamos estudando o PIC18F1220 escolha neste caso este microcontrolador. Em *device flags* são ajustados os configurations bits do microcontrolador. Não vamos nos preocupar sobre estes agora, pressione o botão *default* desta janela para que o mesmo seja configurado desta forma. Agora podemos clicar sobre o botão *ok* e prosseguir com o projeto. Assim que você pressionar o *ok*, a tela da figura 4 surgirá.

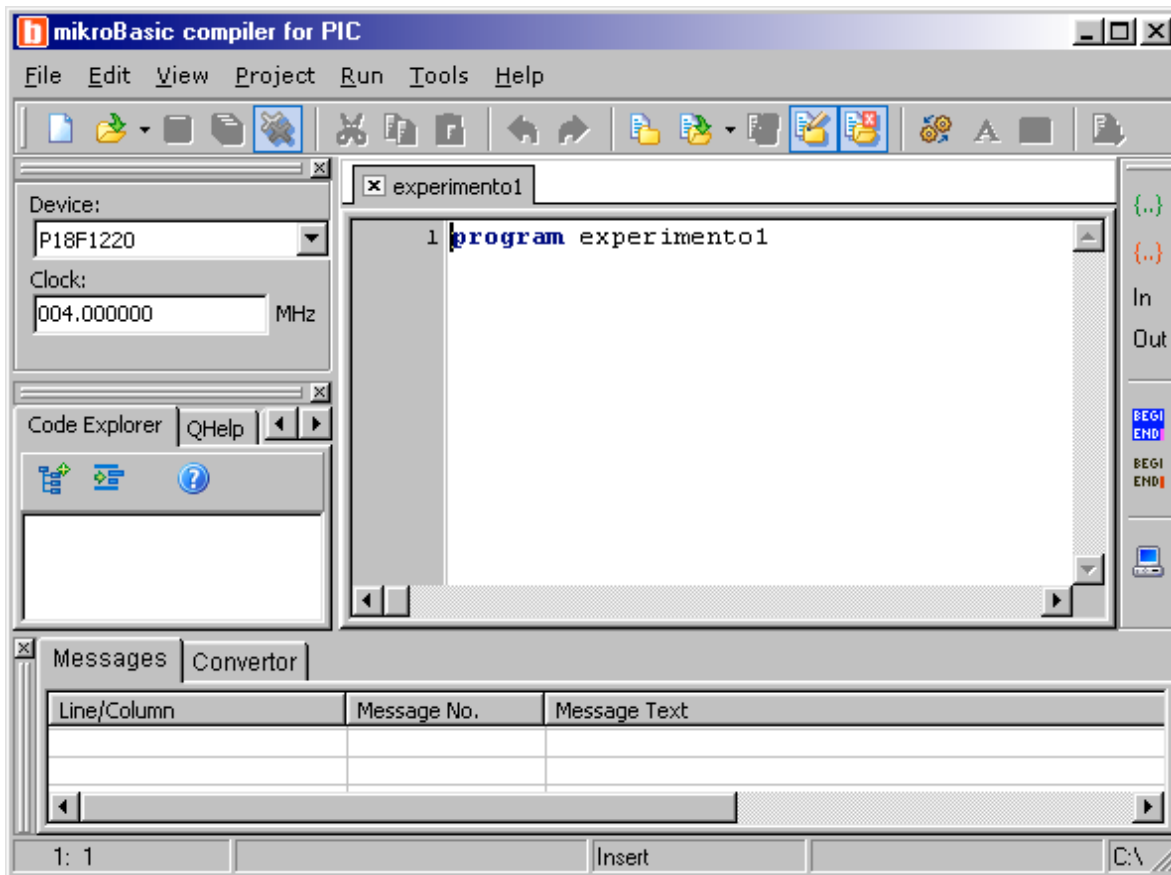


Figura 4 – Tela apresentada após a criação do projeto

Note que a parte branca que está apresentada é onde você deverá criar o seu programa. Um programa em BASIC segue a forma apresentada na tabela 3.

```

program nome_do_projeto

definição de entradas

definição de saídas

definição de variáveis

main:

end.

```

Tabela 3 – Organização de um programa em BASIC

Em *definições de entrada e definições de saída* são informados quais pinos do microcontrolador são utilizados para funções de I/O (entrada e saída). Já em *definições de variáveis* podemos declarar as variáveis que venham a ser utilizados pelo programa. A partir do *main*: começamos a escrever o programa que irá “rodar” no microcontrolador. Finalmente, o programa é finalizado através do *end*.

Após desenvolver todo o programa da forma apresentada, será necessário *compilar* o seu projeto. Compilar significa traduzir o código que está em BASIC para um código que o microcontrolador consiga entender. Para iniciar a compilação, pressione junto CTRL+ F9 ou vá no menu *Project -> Build*. Neste momento inicia-se a compilação e caso haja algum erro, o mikroBASIC solicitará o ajuste do mesmo para iniciar uma nova etapa de compilação.

Primeiro exemplo prático no mikroBASIC

Muito bem, neste ponto já temos subsídios suficientes para desenvolver o nosso primeiro projeto no PIC. Para isso, a placa didática PICLAB18F1220 deve estar conectada ao gravador GPPIC PRO (Desenvolvido pela Cerne Tecnologia) e este deve estar conectado a porta paralela do PC. Este primeiro exemplo consistirá em ler o estado de um botão e dependendo do seu estado, acionar ou não um led conectado a uma saída do microcontrolador. O esquema elétrico deste projeto pode ser verificado na figura 5.

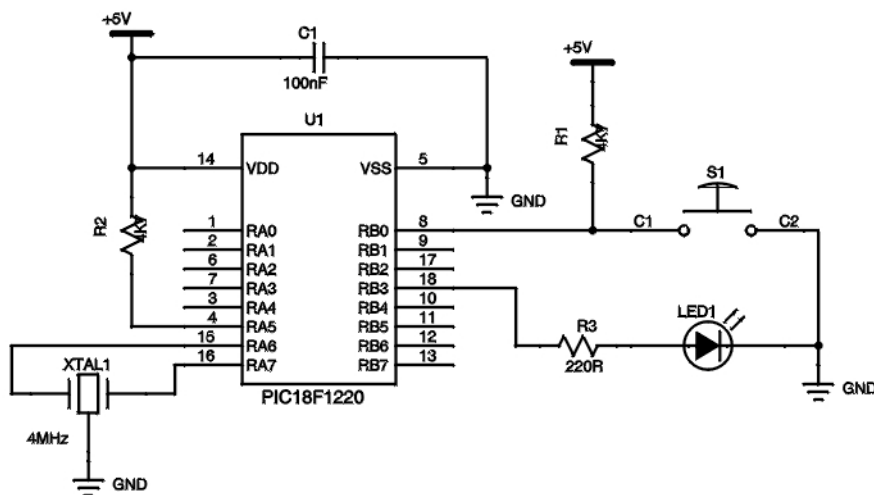


Figura 5 – Esquema Elétrico do exemplo

Observe que o microcontrolador é alimentado em 5 V através dos pinos 5 e 14. O botão está conectado em lógica negativa ao pino RB0. Isto quer dizer que quando o botão está solto, o nível presente na entrada RB0 é 1 e quando o mesmo fica pressionado, este nível vai a 0. O led está conectado ao pino RB3 e este já é acionado em lógica positiva. O ressonador conectado aos pinos RA6 e RA7 geram a freqüência para funcionamento interno do microcontrolador. Note

que no pino RA5 existe um resistor ligado a VCC. Este resistor é chamado de pull up e sua função é garantir o nível alto na entrada RA5, evitando desta forma que o microcontrolador venha a resetar. Como nos PCs, o PIC também a sua entrada de reset que fica no pino RA5. Quando este pino é levado a nível lógico 0, o microcontrolador é resetado e quando fica em 1 o mesmo opera normalmente. Neste exemplo onde a função de reset não é utilizada, este pino foi mantido em nível alto.

O programa que fará o efeito desejado está apresentado na tabela 4.

```
program experimento1

main:
    trisb.0=1
    trisb.3=0

    if portb.0 = 0 then
        portb.3=1
    else
        portb.3=0
    end if

    goto main

end.
```

Tabela 4 – Programa do exemplo

Agora vamos entender melhor como o programa funciona. A linha *program experimento1* informa o nome do projeto definido. Logo em seguida temos o label *main:* onde a partir deste ponto poderemos criar o programa. O PORTB do microcontrolador possui 8 pinos que são chamados de pinos de I/O (entrada e saída). A definição destes pinos, ou seja se eles serão de entrada e saída é feita no registrador TRISB. O bit 0 do registrador TRISB é referente ao RB0, enquanto o bit 1 ao RB1 e assim sucessivamente. Como no RB0 está ligado um botão que para o sistema é uma entrada, este pino foi configurado como entrada através do comando *trisb.0=1*. Quando algum destes bits recebe 1, significa que ele está configurado como entrada e quando recebe 0, como saída. Note que na linha abaixo está definido o *trisb.3* como saída, pois ele recebe 0. Definido a direção dos pinos, é necessário testar o PORTB para saber o estado atual deles. Isto é feito através da declaração *if*. Observe o trecho *if portb.0 = 0 then*. A tradução literal ficaria *se o portb.0 está em nível baixo então*. Ele testa o nível lógico presente na entrada RB0 e se o botão estiver pressionado, este teste será verdadeiro e o comando associado ao *if* será executado que neste caso é para ligar o led através de *portb.3=1*. Caso o botão não esteja pressionado, o *else* (*Senão*) será tratado, deixando neste caso o led desligado através do comando *portb.3=0*. Finalmente o *if* é encerrado através do *end if* (*fim do se*). Este bloco fica se repetindo continuamente, pois logo em seguida observamos o uso do *goto main* que faz com que o programa volte para o label *main* e fique testando o

estado do botão e tomando determinada ação caso o botão esteja pressionado ou não. O programa agora é encerrado através de end.

Após a digitação deste código no mikroBASIC, compile o mesmo e neste ponto poderemos transferir o programa para a placa didática. Existem vários softwares que podem ser utilizados para este fim. Neste exemplo, o software ic-prog foi utilizado. Este software pode ser baixado gratuitamente no site do desenvolvedor, que é www.ic-prog.com. Após o download deste software, inicialize o mesmo. Surgirá uma janela solicitando que você escolha o tipo de gravador utilizado. Deixe esta janela conforme apresentado na figura 6.

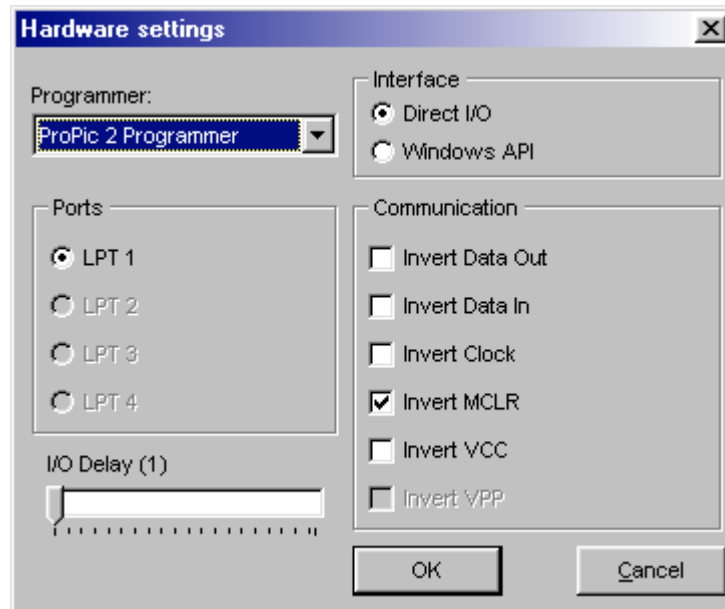


Figura 6 – Configurando o ProPic II

obs: Caso o seu Windows seja o Xp, 2000 ou NT, será necessário liberar o acesso a porta paralela do seu PC. Vá no site da Cerne Tecnologia (www.cerne-tec.com.br) e veja na seção tutoriais o guia Liberando o acesso da porta paralela.

Após esta configuração, pressione o botão *ok*. Será aberto agora a janela normal do Ic-Prog, como apresentado na figura 7.

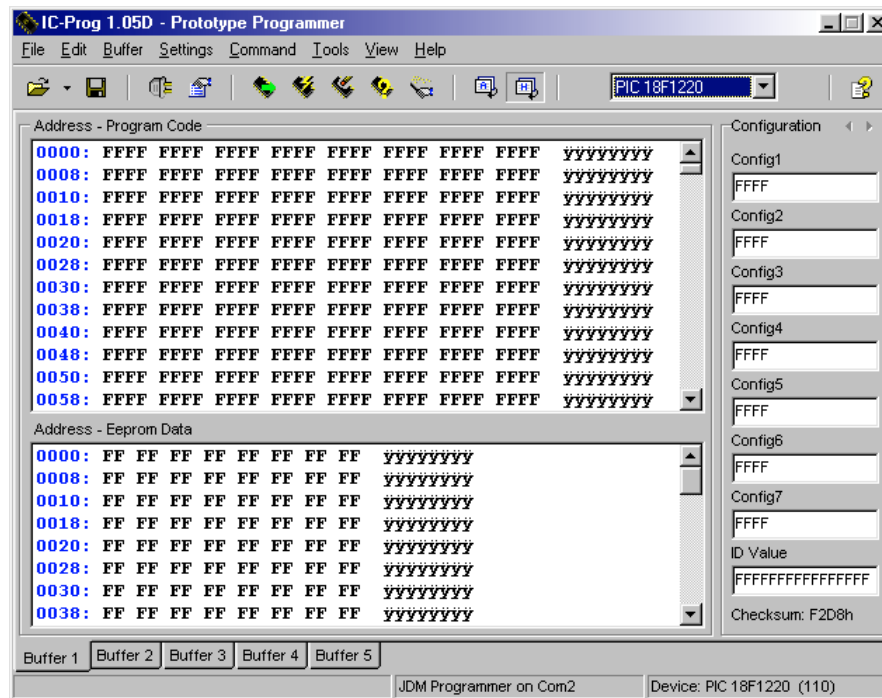


Figura 7 – Janela do Ic-Prog

Será necessário agora escolher o microcontrolador utilizado para gravação, neste caso o PIC18F1220. Para isso vá em *Settings* -> *Device* -> *Microchip PIC* e procure na lista apresentada o microcontrolador PIC18F1220.

Podemos também trocar o idioma utilizado pelo Ic-Prog para o Português. Vá em *Settings* -> *Options* e abra a aba *Language*. Escolha no combo presente neste aba, o idioma *Portuguese* e pressione *ok*.

Agora podemos abrir o arquivo a ser gravado no microcontrolador. Sempre que o mikroBASIC compila um arquivo e não houve nenhum erro de compilação, o compilador gera um arquivo com o mesmo nome do projeto porém com a extensão hex. É este arquivo que deve ser gravado no microcontrolador. Para abri-lo, vá em *Arquivo* -> *Abrir* e veja na pasta onde você criou o seu projeto o arquivo com a extensão hex. Clique duas vezes sobre o mesmo e pressione a tecla F5. Neste momento uma janela como a apresentada na figura 8 surgirá.

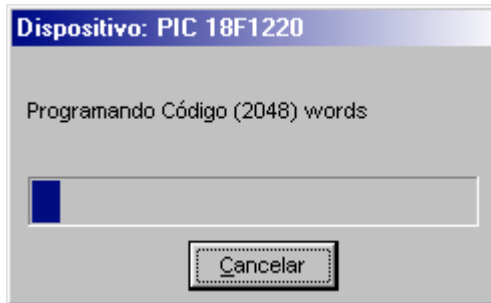


Figura 8 – Gravação do microcontrolador

Após a gravação, teste o seu programa recém gravado na placa didática. Pressione o botão que fica ligado ao RB0 e veja se o led acende de acordo com o pressionar dele.

Este foi o primeiro de uma série de artigos que serão apresentados com o mikroBASIC e o PIC. Dúvidas, sugestões ou críticas o meu e-mail de contato é vitor@cerne-tec.com.br.