TEEE I-Projeto de Robôs Móveis

MICROCONTROLADORES -PIC

1. INTRODUÇÃO

Ao longo dos anos observa-se que o interesse pelo domínio do uso dos microprocessadores/microcontroladores tem sido constante, pois com eles pode-se desenvolver uma variedade de aplicações e circuitos de grande utilidade, automatizando os sistemas encontrados no mundo moderno.

Os microcontroladores são fortemente usados em circuitos de automóveis, eletrodomésticos, brinquedos, na robótica e aparelhos de um modo geral que necessitam de automação e controle.

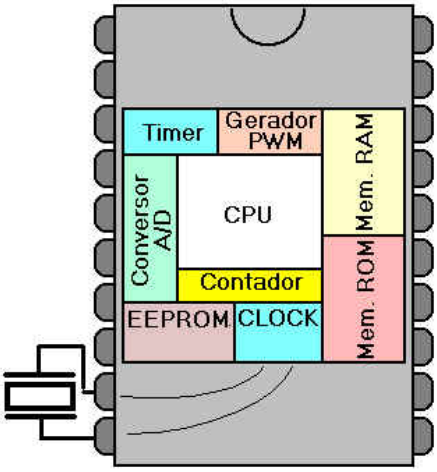
O mercado de componentes eletrônicos oferece várias opções de fabricantes e diferentes dispositivos microcontroladores com características próprias, além disso, existem as placas comerciais prontas para o uso e de custo acessível, sem a necessidade de desenvolver a sua própria ou um circuito, bastando aprender a usá-la e a sua programação.

Os Microcontroladores surgiram no princípio dos anos 80. Trata-se de um circuito integrado programável que contém toda a estrutura (arquitetura) de um microcomputador, isto é, dentro de um microcontrolador podemos encontrar uma CPU (Unidade Central de Processamento), memória RAM, memória EEPROM (Memória de leitura e escrita não volátil; portas de entrada/saída (Pinos de E/S). Inclusive muitos modelos de microcontroladores incorporam diferentes módulos

"periféricos", como conversores analógico/digital (A/D) , módulos PWM (controle por largura de pulso), módulos de comunicação serial ou paralelo e muito mais.

Todos esses recursos são encontrados dentro do mesmo circuito integrado. Mas, a grande diferença com relação aos microprocessadores está na **pequena capacidade de suas memórias e na velocidade de processamento.**

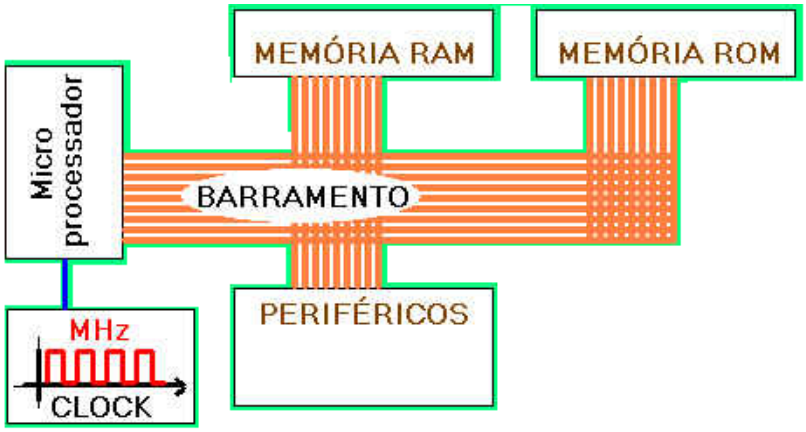
Figura 1- Ilustrando o Microcontrolador e os seus recursos .



Fazendo um paralelo, um sistema microprocessado deve possuir:

- Um **microprocessador** (CPU)
- **Memória ROM** (onde fica gravado o programa principal)
- **Memória RAM** (utilizada como memória de rascunho pelo programa executado)
- **Periféricos** - Entrada e Saída (são os dispositivos conectados ao sistema, geralmente sensores e atuadores.)
- **Barramento** - fios ou trilhas condutoras dos pulsos elétricos que permitem a interconexão dos componentes.
- **Circuito gerador de CLOCK** - circuito que gera um sinal cíclico em uma determinada freqüência de tempo.

Figura 2- Ilustrando o Microprocessador e barramentos



Como podemos observar mesmo um **sistema microprocessado** possui uma série de componentes, no caso dispostos em vários CIs e interligados por condutores paralelos que constituem o **barramento**.

Os produtos que incorporam microcontroladores em seu sistema visam principalmente, aumentar seus recursos, reduzir seu tamanho e custo, melhorar sua confiabilidade e diminuir o consumo de energia. Alguns dos fabricantes de microcontroladores atuais são a Intel, Microchip, Atmel, Motorola, etc.

Alguns dos microcontroladores de baixo custo usados no mundo da automação são:

1. **Intel 8051** e suas variações como o 8031. Possui um assembly relativamente simples, e baixo custo. Bastante conhecido e muito utilizado com fins didáticos. Inspirou a construção de diversos outros modelos similares, com mesmas instruções e capacidades, no entanto fabricado por outros fabricantes, como Atmel, Zilog, etc.
2. **PIC16F84 (Microchip)** surgiu como um dos mais utilizados microcontroladores pelos "hobistas" pela sua versatilidade, facilidade de programação e baixo custo. Hoje, a família de microcontroladores PIC é muito utilizada em projetos de automação.
3. **Microcontroladores ATMEL**, de excelente custo.
4. **Microcontroladores COP**
5. **Microcotroladores HC05, HC08 e HC11** da **FREESCALE (MOTOROLA)** também são bastante utilizados, apresentando como vantagem o seu baixo custo.

Um dos microcontroladores bastante procurado para desenvolvimento de projetos é o "PIC" do fabricante "Microchip", onde " PIC" significa "*Peripheral Interface Controller"*

Existe uma grande quantidade de microcontroladores PIC, cujas características e recursos variam de um modelo para outro. Assim sendo, os projetistas podem selecionar o modelo que melhor se ajuste as suas necessidades. Os diferentes modelos de microcontroladores se agrupam por "família". Uma família pode ser formada por um conjunto de modelos cujas características e recursos são bastante similares.

Um microcontrolador precisa ser programado, ou seja, o projetista deve escrever um programa que contenha todo o processamento que o microcontrolador deve executar. Este programa pode ser escrito em uma linguagem chamada "Assembly" que por se tratar de uma linguagem de "baixo nível" e se encontrar "mais próximo" à linguagem da máquina (binária) apresenta uma relativa complexidade. Por isso, a realização de projetos com esta linguagem exige um grande esforço intelectual e muito mais tempo de projeto.

Utiliza-se neste curso o PIC16f877 por ter literatura acessível, fácil de ser programado e atender aos objetivos do curso. Este microcontrolador apresenta diversos recursos já embutidos, dos quais podemos citar: 8 entradas analógicas, saídas PWM, 8kBytes de memória ROM e 33 I/Os.

2. PIC 16F877A- Características Gerais

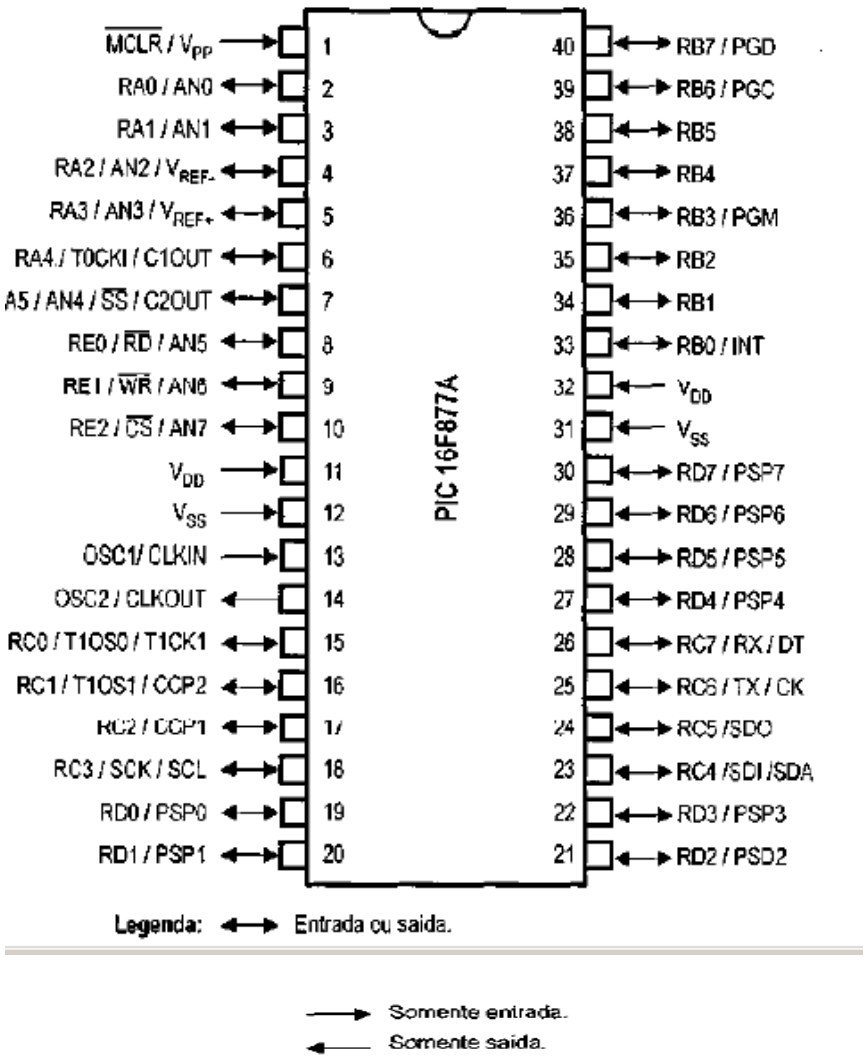
O dispositivo programável PIC 16F877A, possui várias funções na sua arquitetura , muito importantes em projetos de automação, como por exemplo:

- Microcontrolador com 40 pinos, incluindo pinos analógicos e 33 I/O's;
- 15 tipos de interrupções disponíveis: *timers*, contagem, pulso externo, etc;
- *Clock* de 4 até 20MHz;
- Pilha com oito níveis de profundidade;
- Memória RAM com 368 *bytes*;
- Conversores Analógico/Digital (A/D);
- Modo *Sleep*;
- Diferentes opções para osciladores (*crystal*, oscilador RC, etc);
- PWM;
- Modo USART (para comunicação serial);
- 8kBytes de memória ROM;

2.1 - Pinagem e funções

Mostra-se na figura 3 a pinagem e um resumo da descrição e função de cada pino deste microcontrolador. Deve ser notado que a maioria dos pinos possui várias funções, não significando que as funções possam ser exercidas ao mesmo tempo.

Figura 3- Pinagem DIP-Microcontrolador PIC 16F877A.



1. MCLR : Master Clear – Quando em nível baixo (0V), define situação de RESET. Quando em nível alto (5V), determina programa em execução. VPP: Tensão de programação – Quando este pino estiver em 13.4V, o microcontrolador entra em modo gravação, permitindo a transferência de um programa via ICSP;

1. RA0 **–** Entrada / saída digital. AN0 – Entrada analógica canal 0 para o ADC interno;
2. RA1 **–** Entrada / saída digital. AN1 – Entrada analógica canal 1 para o ADC interno;
3. RA2 – Entrada / saída digital. AN2 – Entrada analógica canal 2 para o ADC interno. Vref- – Uso do pino para definir a referência negativa para o conversor AD;

4. RA3 – Entrada / saída digital. AN3 – Entrada analógica canal 3 para o ADC interno. Vref+ – Uso do pino para definir a referência positiva para o conversor AD;

5. RA4 – Entrada / saída digital. TOCKI – Contador rápido;

6. RA5 – Entrada / saída digital. AN4 – Entrada analógica canal 4. SS – *Slave Select* para porta serial síncrona;

7. RE0 – Entrada / saída digital. RD – Entrada de controle de leitura para porta paralela escrava. AN5 – Entrada analógica canal 5;

8. RE1 – Entrada / saída digital. WR – Entrada de controle de gravação para porta paralela escrava. AN6 - Entrada analógica canal 6;

9. RE2 – Entrada / saída digital. CS – “Chip Select” para porta paralela escrava. AN7 – Entrada analógica canal 7;

10. VDD – Alimentação (preferência 3V a 5V);

11. VSS – Referência (0V / GND);

12. OSC1/CLKIN – Pino para ligação do circuito oscilador externo (entrada). Usado em conjunto com o pino OSC/CLKOUT. Recomendado usar cristal de 4 a 20 MHz;

13. OSC2/CLKOUT – Pino para ligação do circuito oscilador externo (saída);

14. RC0 – Entrada / saída digital. T1OSO – Saída do oscilador do TIMER1. T1CKI – Entrada de clock para TIMER1;

15. RC1 – Entrada / saída digital. T1OSI – Entrada do oscilador do TIMER1. CCP2 – Entrada de captura 2, saída de comparador 2 ou PWM 2;

16. RC2 – Entrada / saída digital. CCP1 – Entrada de captura 1, saída de comparador 1 ou PWM 1;

17. RC3 – Entrada / saída digital. SCK/SCL – Entrada ou saída de sinal de clock serial síncrono para SPI e I2C;

18. RD0 – Entrada / saída digital. PSP0 – Pino 0 da porta paralela escrava;
19. RD1 – Entrada / saída digital. PSP1 – Pino 1 da porta paralela escrava;
20. RD2 – Entrada / saída digital. PSP2 – Pino 2 da porta paralela escrava;
21. RD3 – Entrada / saída digital. PSP3 – Pino 3 da porta paralela escrava;
22. RC4 – Entrada / saída digital. SDI – Entrada de dados em SPI. DAS – Entrada/saída de dados em modo I2C;
23. RC5 – Entrada / saída digital. SD0 – Saída de dados SPI;
24. RC6 – Entrada / saída digital. TX – Pino para transmissão serial assíncrona. CK – *Clock* para transmissão síncrona;
25. RC7 – Entrada / saída digital. RX – Pino para recepção serial assíncrona. DT – Dados da serial síncrona;
26. RD4 – Entrada / saída digital. PSP4 – Pino 4 da porta paralela escrava;
27. RD5 – Entrada / saída digital. PSP5 – Pino 5 da porta paralela escrava;
28. RD6 – Entrada / saída digital. PSP6 – Pino 6 da porta paralela escrava;
29. RD7 – Entrada / saída digital. PSP7 – Pino 7 da porta paralela escrava;
30. VSS - Referência (0V / GND);
31. VDD – Tensão de alimentação (mesma que pino 11);
32. RB0 – Entrada / saída digital. INT – Entrada de sinal de interrupção via hardware;
33. RB1 – Entrada / saída digital;
34. RB2 - Entrada / saída digital;
35. RB3 – Entrada / saída digital. PGM – Entrada de sinal para gravação em baixa tensão (5V);
36. RB4 – Entrada / saída digital;

37. RB5 – Entrada / saída digital;

38. RB6 – Entrada / saída digital. PGC – Clock para programação ICSP ou pino para depuração;

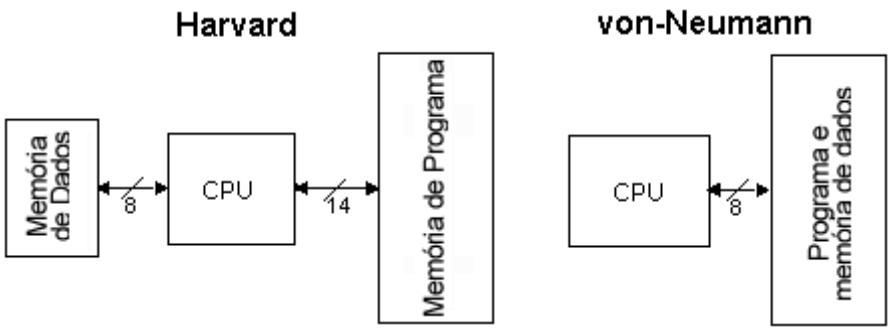
39. RB7 – Entrada / saída digital. PGD – Dados para programação ICSP ou pino para depuração;

2.2 - Arquitetura interna e funcionamento

A arquitetura de trabalho do PIC é a ***Havard***, Figura 4, que utiliza dois barramentos em seu funcionamento, um para dados que normalmente possui 8 bits e outro de endereçamento de memória, com 14 bits, usualmente. Nessa arquitetura, tem-se uma velocidade de processo maior, pois uma vez utilizado um endereçamento, enquanto a CPU trabalha com os dados, o barramento de endereços procura e coloca a posição do próximo comando à disposição da CPU.

No modo de operação ***Von-Neumann*** é necessário mais tempo e comandos de memória para o funcionamento, pois os dados e os endereços necessitam "dividir" o mesmo barramento, como mostrado na Figura 4.

Figura 4- Arquitetura *Havard* x V*on-Neumann*.



Na Figura 5 ilustra-se a Arquitetura Interna, com os vários componentes de hardware. Nesta figura pode-se observar a presença de 5 Ports de I/Os para comunicação externa, entre os quais, tem-se 8 pinos para entrada/saída de sinais analógicos.

O barramento de dados se encontra à direita do contador de programa (*PC-Program Counter*) que está diretamente ligado à memória FLASH de programa. O barramento de programa está diretamente associado ao conjunto de

instruções do PIC e, por conseguinte, este se encontra multiplexado para dar acesso à ULA (Unidade Lógica e Aritmética).

Para executar uma operação (lógica /aritmética) utiliza-se a ULA, que decodifica o tipo de instrução a ser realizada primeiramente e, logo depois, executa a operação nos dados que virão posteriormente com ajuda do MUX (Multiplexador). O resultado é guardado no **WReg** que é um dos registradores mais importantes do PIC (registrador acumulador).

Do lado esquerdo da ULA, têm-se os registradores que provém das configurações iniciais do PIC. Estas configurações precisam ser especificadas no início de qualquer programa, tais como,

-***Brown-Out Reset***: tem a função de esperar a alimentação se estabilizar antes de fornecer energia ao microcontrolador;

- ***Oscillator Start-up Timer***: o programador deve especificar que tipo de *clock* está usando: crystal, oscilador RC, etc;

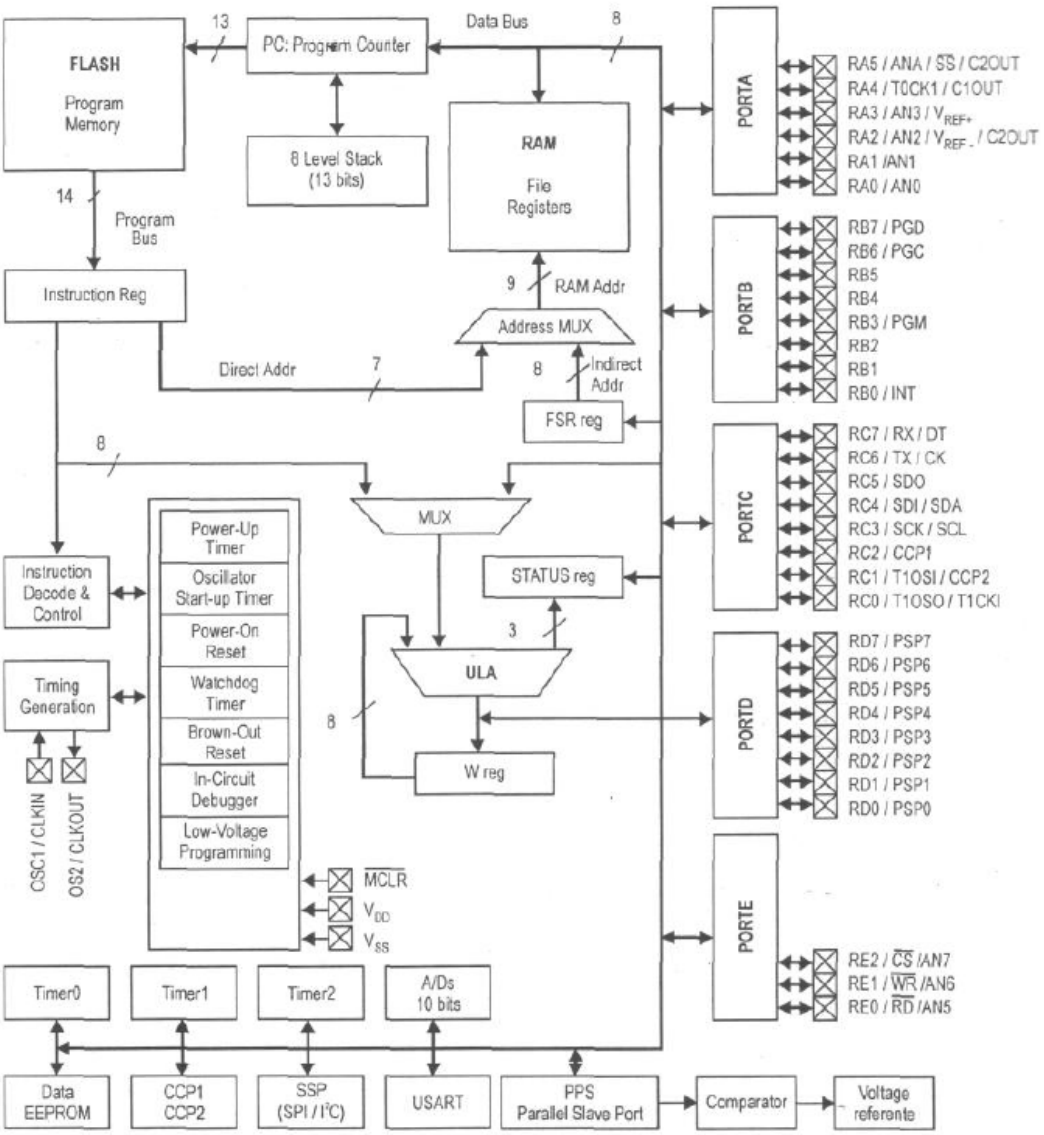
- ***Watchdog Timer***: em prevenção a erros no programa, este é um circuito de *reset* automático do PIC a um determinado tempo (de 0 a 255), que pode ser ou não habilitado na programação;

- ***Low-Voltage Programming***: circuito que nos auxilia na programação do microcontrolador;

- ***Instruction Decode & Control***: registrador que possui os códigos de máquina que serão utilizados pela ULA para cada instrução feita pelo programador. A função desse registrador é interpretar a linguagem de programação *Assembly* e decodificá-la em linguagem de máquina.

Na Figura 5, tem-se mais algumas funções do PIC, destacando o modo USART que é uma comunicação externa que o PIC possui para através de uma porta RS232 se comunicar com um computador. Também se encontram timers, modo comparador tensão de referência, conversor digital analógico (por causa das entradas analógicas do PIC), entre outros.

Figura 5- Arquitetura interna do PIC.



2.3 – Memórias de Programa x Memória de Dados

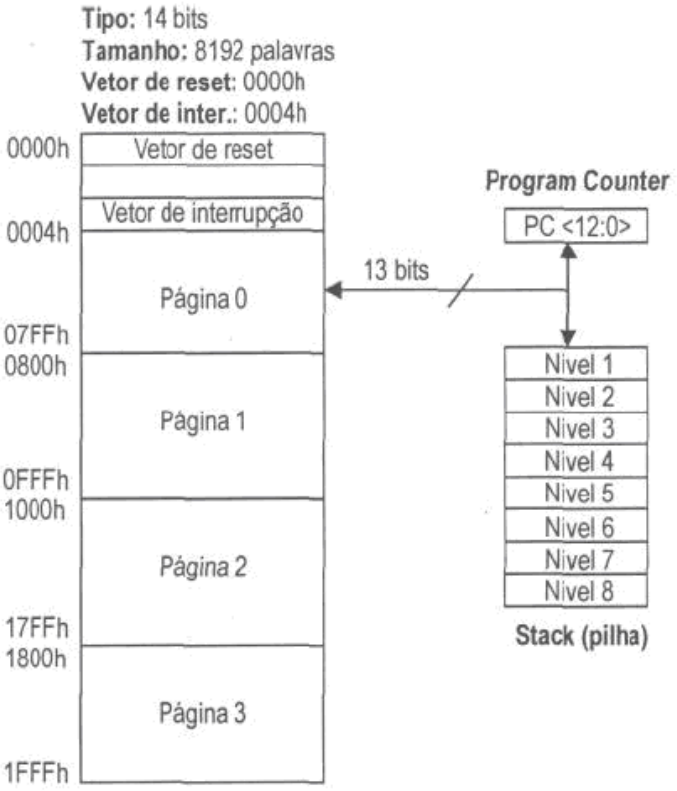
Memórias de Programa- O PIC possui uma memória de programa de 14 bits do tipo **FLASH** (Figura 6), regravável eletronicamente e de grande velocidade de trabalho. Essa memória é designada para a programação do PIC, possuindo as seguintes características:

Vetor de reset: onde o programa inicia (PIC 16F877A), endereço 0x00H.

Vetor de interrupção: após cada interrupção o contador de programa (PC) pula automaticamente para essa posição que, no PIC 16F877A, possui o endereço 0x 04H.

Pilha: a pilha guarda o endereço para os desvios de fluxo na programação. O PIC 16F877A possui uma pilha de 8 níveis, ou seja, ela pode guardar até 8 endereços de retorno.

Figura 6- Mapa da Memória de Programa-FLASH.



Memória de Dados - a memória de dados é dividida em 4 bancos de memória para facilitar a programação. Essa memória é constituída de vários registradores

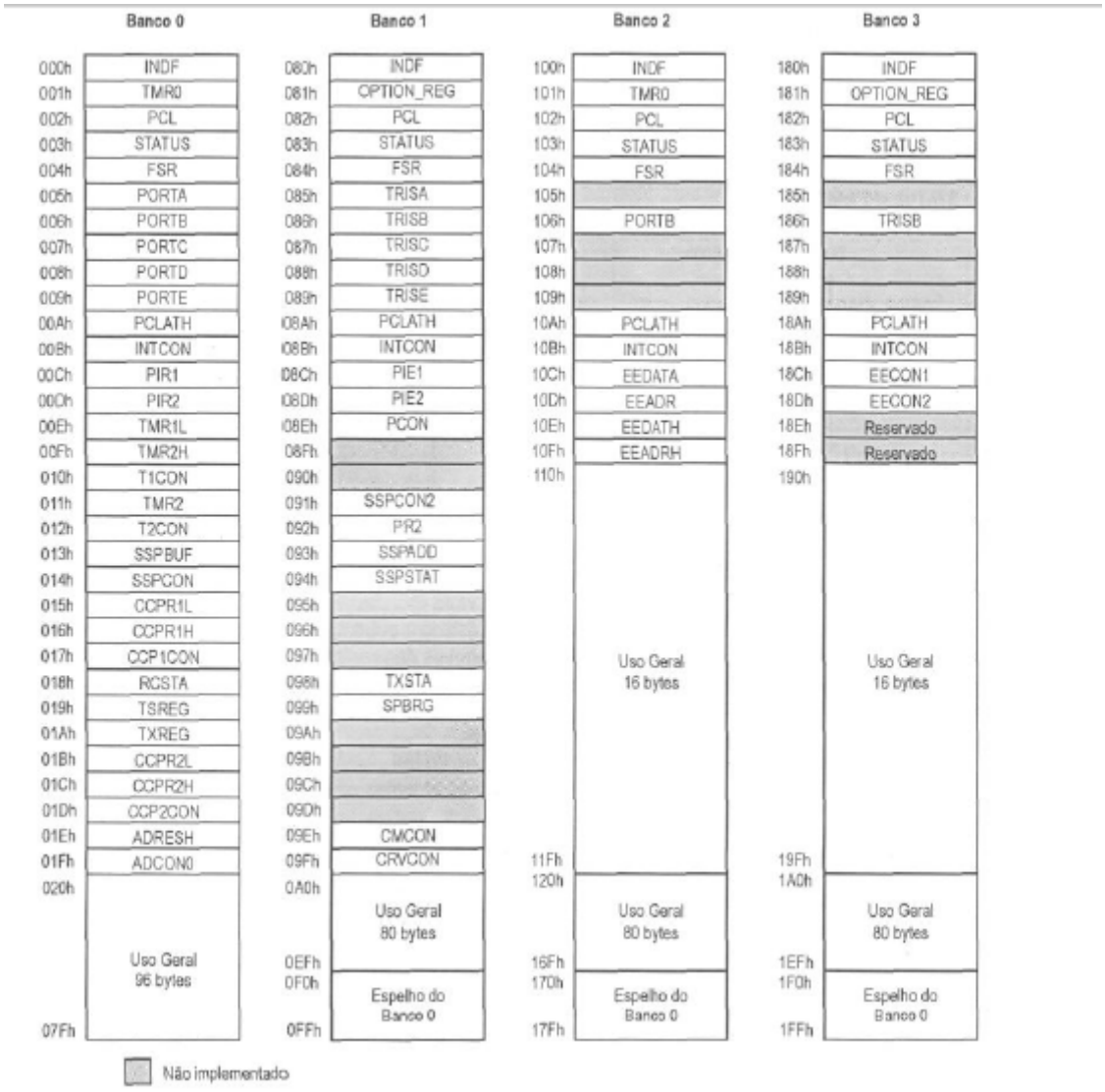
para guardar os dados de 8 bits a serem usados na programação e configuração de trabalho do microcontrolador. Existem dois tipos de registradores:

Registradores Especiais (SFR's): são pré-determinados pelo fabricante. Normalmente possuem locais para que se possa configurar o modo de operação do microcontrolador. Por exemplo: para definir se o PORTB vai ser entrada ou saída, tem-se o registrador especial TRISB.

Registradores de Uso Geral: são registradores definidos pelo usuário. Fazendo-se uma comparação, é o mesmo que as variáveis criadas em linguagens de programação, com a diferença que se nomeia o endereço do dado e não o dado em si.

Observa-se na Figura 7, o mapa da **Memória de dados** para o PIC 16F877A,

Figura 7- Bancos de memórias para os registradores de dados.



3. CIRCUITOS EXTERNOS PARA O PIC.

Para o uso do PIC são necessários alguns circuitos auxiliares que fazem com que o microcontrolador funcione plenamente e exerça suas funções. Entre esses circuitos estão o circuito de *clock*, *reset* e alimentação.

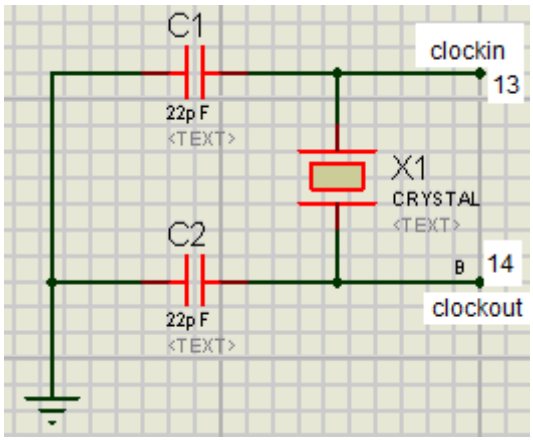
3.1 Circuito de *Clock*

A CPU (*Central Processing Unit*) do PIC necessita de um sinal de sincronismo , *clock,* para funcionar . Um quarto do *clock* gera o que se chama de ciclo de máquina, que fornece sua base de tempo. Por exemplo, uma instrução pode demorar 2 ciclos de máquina . O número de ciclos de máquina necessários para a execução de cada instrução é variado para cada tipo de comando.

$$CK_{INT} = \frac{CK_{EXT}}{4} \qquad (1)$$

O circuito de *clock* externo mais usado em projetos com PIC é que utiliza um *crystal* oscilador, por apresentar uma melhor estabilidade do sinal. O esquema de ligação para esse tipo de *clock* pode ser visto na Figura 8.

Figura 8- Esquema de ligação para o *crystal de quartzo*

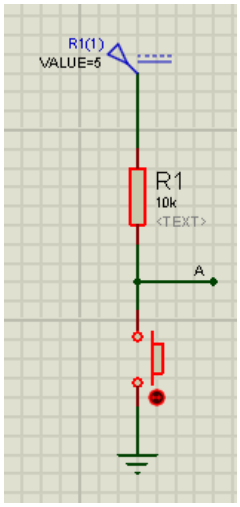


3.2 - Reset

O circuito de *reset* do microcontrolador reinicializa a execução do programa. Um sinal de nível baixo (*low*) no pino 1 ($\overline{MCLR}$ - *Master Clean Reset*),

reseta o microcontrolador . Esse controle é feito por uma chave conforme se mostra no circuito da Figura 9,

Figura 9- Circuito para reset manual do PIC.



O ponto A da figura 9 é ligado no pino 1 do PIC 16F877A. Quando o botão é apertado, o contador de programa (PC) voltará a indicar o endereço 0x00H do programa principal, além de resetar todos os níveis lógicos do *chip*.

4 - *Softwares* Auxiliares

Como dito anteriormente, o microcontrolador é um chip programável. Portanto, necessita de *softwares* auxiliares para o desenvolvimento de um projeto no caso do PIC, podem ser utilizados três *softwares*:

- PROTEUS 7 PROFESSIONAL, para simulação do projeto;
- MIKRO C PRO FOR PIC, para a programação em linguagem C e;
- MPLAB (proprietário do gravador-Microchip) para a gravação do programa no microcontrolador.

4.1 - *Software* de Simulação

Para simular os projetos do PIC, pode-se usar o programa PROTEUS 7 PROFESSIONAL que fornece uma vasta biblioteca de componentes que são muito úteis e fortemente recomendados para ser utilizado como ferramenta de simulação de circuitos, tanto digitais, quanto analógicos.

Esse programa consta de dois aplicativos que são: *ISIS* e o *ARES*. O *ISIS* é a ferramenta de simulação de circuitos e o *ARES* é uma interface para a criação de placas impressas.

4.2 - *Softwares* de Desenvolvimento do Programa.

A programação do PIC pode ser feita em linguagem Assembly usando para isso o programa MPLAB. A linguagem C oferece vantagens, como a compactação e desta forma utiliza menos espaço na memória de programa. Programando em C, um dos programas bastante utilizado é o *software* MIKRO C FOR PIC .

4.3 - *Softwares* de Gravação

O MPLAB também é utilizado para a gravação dos programas no microcontrolador da família PIC, usando o gravador da mosaico produtos.

O aplicativo ISIS permite fazer os esquemas de ligações para o circuito. Caso seja necessário fazer uma placa de circuito impresso, pode-se utilizar o aplicativo ARES. Para isso leva-se todas as ligações feitas no ISIS diretamente para o ARES, clicando no ícone do ARES que aparece na interface do ISIS.