

Arquitectura de Computadores II



3º Ano

Periférico Genérico PG01/04

João Luís Ferreira Sobral
Departamento do Informática
Universidade do Minho

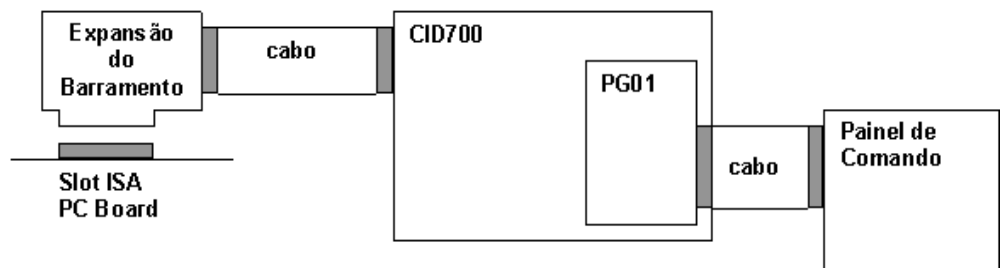


Março 2003

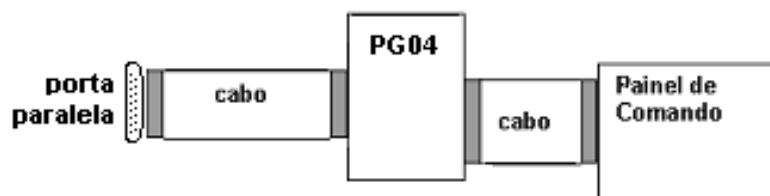
Bancada de Ensaios

● Introdução

- A bancada de ensaios destina-se a apoiar as aulas laboratoriais de estudo da interface do micro-computador PC compatível com o exterior
- A bancada de ensaios é constituída por dois elementos:
 - Periférico genérico - inclui portas de entrada e de saída, um conversor A/D e um conversor D/A
 - Painel de comando - inclui um visor de 7 segmentos, 8 interruptores, 8 *leds*, um microfone e uma fonte de tensão regulável
- Existem duas versões do periférico genérico:
 - PG01 – Versão para o barramento ISA de 8 bits que encaixa num sistema de expansão do barramento designado CID700 que facilita o acesso aos sinais do barramento e a sua protecção



- PG04 – Versão para a porta paralela

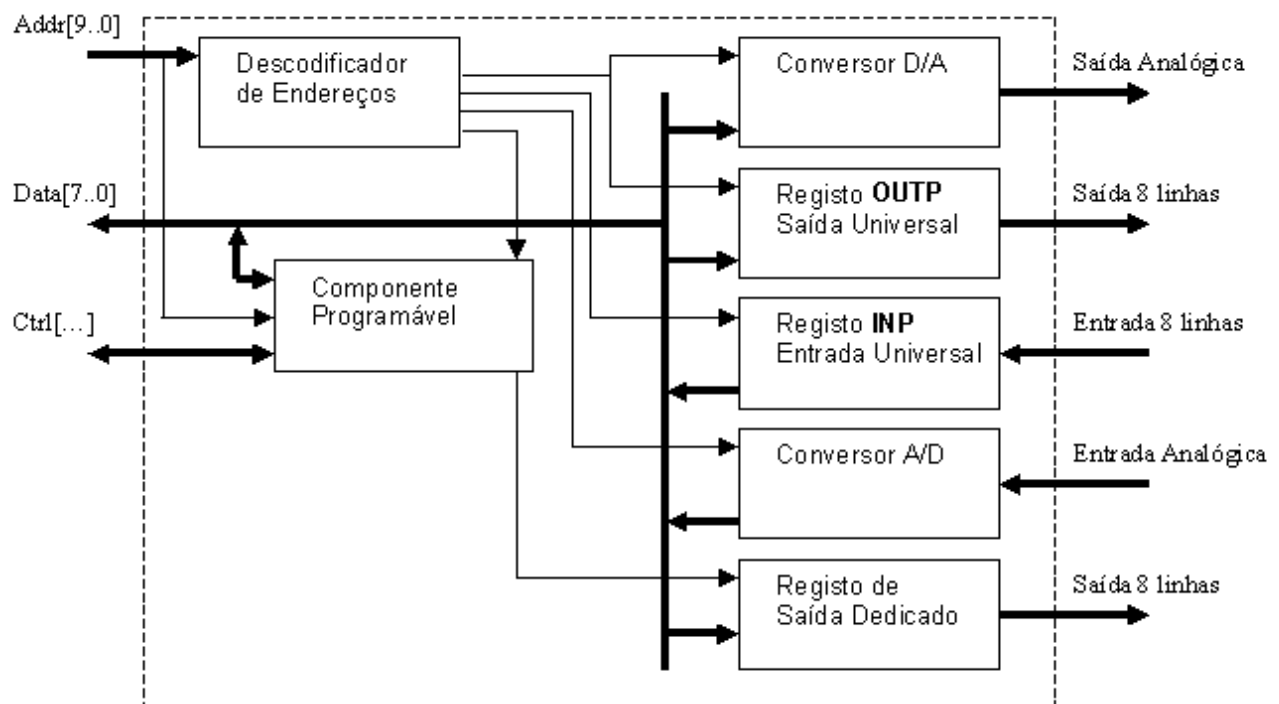


Periférico genérico

● Características

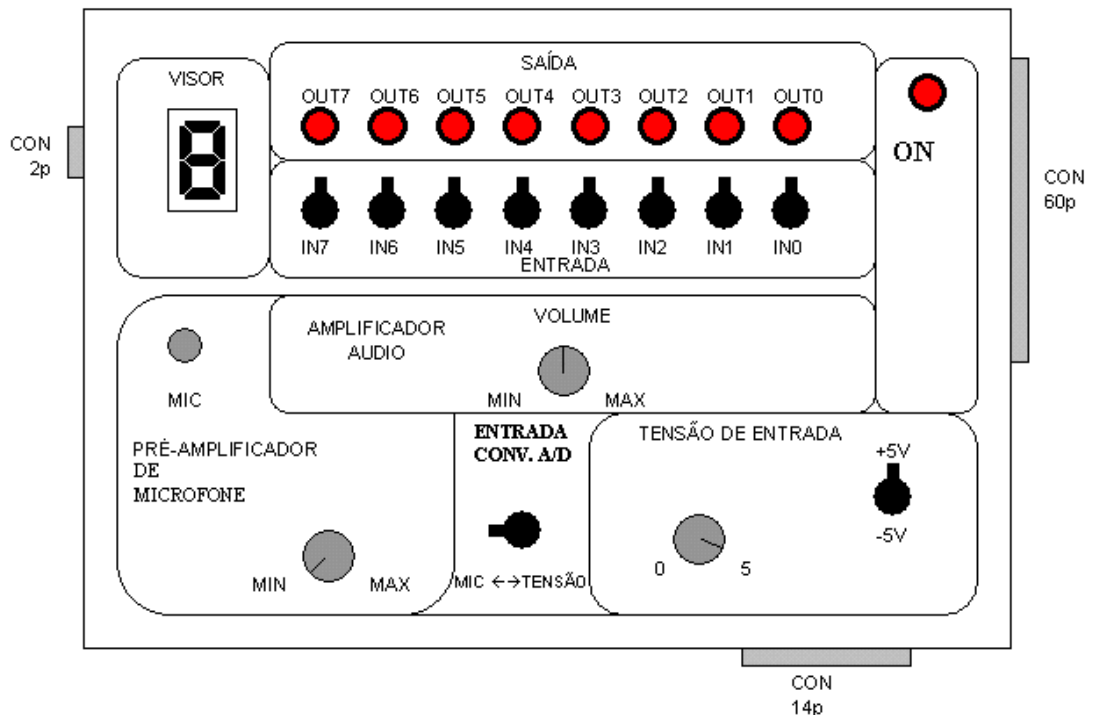
- Uma porta de saída universal de 8 bits (OUTP);
- Uma porta de entrada universal de 8 bits (INP);
- Uma porta de saída dedicada a um visor de 7 segmentos com ponto decimal;
- Um conversor Analógico/Digital (A/D) de 8 bits;
- Um conversor Digital/Analógico (D/A) de 8 bits;
- Um módulo programável, com 11 pinos de E/S e um conjunto de ligações ao barramento, pré-definidas, de modo a poder utilizar a interrupção IRQ3 e o canal 3 de DMA (apenas PG01).

● Diagrama de blocos



- Os sinais do lado esquerdo são recolhidos do sistema de desenvolvimento CID700 ou da porta paralela. Os sinais do lado direito do diagrama são ligados por cabo ao Painel de Comando.

Painel de Comando



Características

- **8 visualizadores** vulgarmente conhecidos por *leds*
- **8 interruptores**
- **visor de 7-segmentos** com ponto decimal.
- fonte de tensão regulável de $-5V$ a $0V$ e de $0V$ a $+5V$. Possui dois comandos: um **potenciômetro** para variar a tensão e um **comutador** para seleccionar tensão positiva ($0V$ a $+5V$) ou tensão negativa ($-5V$ a $0V$);
- microfone ligado a um pré-amplificador com ganho variável, regulável por um **potenciômetro**;
- amplificador de saída para ligar a um altifalante com controlo de volume por um **potenciômetro**;
- **comutador** que selecciona a entrada analógica entre a fonte de tensão e o pré-amplificador do microfone;

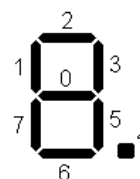
Ligação periférico – painel de comando

Periférico genérico	Painel de comando
OUTP	<i>leds</i>
INP	interruptores
Registo dedicado	visor 7 segmentos
Entrada analógica	fonte de tensão/microfone
Saída analógica	amplificador

Periférico genérico PG01 (Barramento ISA)

● Modo de funcionamento

- As portas de dados podem ser escritas e lidas directamente (com inp e outp)
- O conversor A/D é comandado pelo bit 0 do endereço 0x304. Logo que é colocado o valor '1' neste bit é iniciada uma nova conversão. Quando o conversor acaba de efectuar uma conversão assiná-la com um '0' no bit0 do endereço 0x300.
- Esquema de bits do visor de 7 segmentos:



● Portas de E/S

Endereço E/S	LEITURA	ESCRITA
0x300	ESTADO: Bit 0 activo a '0' indica o fim da conversão A/D	DADOS: Valor a colocar no visor de 7-segmentos
0x304	DADOS: Valor do conversor A/D	COMANDO: Bit 0 activo a '1' inicia a conversão A/D
0x308		DADOS: Valor a colocar na saída OUTF (leds) Valor para o conversor D/A
0x30C	DADOS: Valor na entrada INP (interruptores)	

Periférico genérico PG04 (Porta Paralela)

• Modo de funcionamento

1. Indicar na porta de controlo (bits 5-0) a funcionalidade pretendida
 2. Escrever ou ler na porta de dados (ou estado) os valores pretendidos
- A porta paralela deve ser colocada em modo SPP bidireccional. Para tal deve ser escrito o valor 0x35 no endereço (0x378+0x402)
 - A conversão AD é iniciada enviando o valor 0x02 para a porta de comando, seguido de 0x00. O fim da conversão é assinalado em 0x379
 - Esquema de bits do visor de 7 segmentos:



• Portas de E/S

Endereço E/S	LEITURA	ESCRITA
0x378	DADOS: Valor na entrada INP (interruptores) Valor do conversor A/D	DADOS: Valor a colocar na saída OUTP (leds) Valor a colocar no visor de 7-segmentos Valor para o conversor D/A
0x379	ESTADO: Bit 6 activo a '0' indica o fim da conversão A/D	
0x37a		COMANDO: Bit 5-0: 0x00 – Activa saída para OUTP (leds) 0x21 – Activa entrada para INP (interr.) 0x02 – Inicia conversão A/D 0x23 – Activa entrada conv. A/D 0x04 – Activa saída para D/A e OUTP 0x05 – Activa saída para visor e OUTP

Exercício 1: Portas de Entrada e de Saída

Objectivos

- Acesso ao espaço de endereços de entrada e saída
- Programação básica de um dispositivo periférico
- Interface em Linguagem C

Exercício

Considere as seguintes ligações do PG para o exterior:

- **registo de saída (outp)**
 - 8 leds
 - bit a 1 - acende
 - bit a 0 - apaga
- **registo de entrada (inp)**
 - 8 interruptores
 - ligado - bit a 1
 - desligado - bit a 0

Escreva os programas que realizam as seguintes operações:

- a) Mostrar o valores dos interruptores, da esquerda para a direita: 1) no ecrã 2) nos *leds*.
- b) Acender e apagar todos os *leds*: 1) ao mesmo tempo 2) um a um, da esquerda para a direita e vice-versa.

Exercício 2: Portas de Controlo/Estado

Objectivos

- Fazer a distinção entre os diversos tipos de registos num periférico
- Comandar um periférico através de técnicas de sondagem

Exercício

Considere que às ligações do PG (no exercício anterior) juntamos:

- **visor**
 - 7 segmentos + ponto
 - bit a 1 - acende traço
 - bit a 0 - apaga traço
- **conversor A/D**
 - escala de -5 a 5 Volt
 - cada unidade 0,04 Volt
- **entrada de tensão**
 - pino conversor (ANA-IN)
 - potenciómetro
 - variação tensão

Utilize as técnicas de controlo de periféricos por sondagem para realizar as seguintes operações:

- a) Dar início à conversão A/D e mostrar o valor obtido no ecrã.
- b) Repetir o exercício, mas desta vez, mostrando a parte inteira do valor no visor de sete segmentos. Use o ponto para assinalar valores negativos.
- c) Determinar quantas conversões A/D são realizadas por segundo.

Exercício 3: Conversores A/D e D/A

Objectivos

- Familiarização com as portas de controlo/estado e as de transferência de dados
- Construção de programas para comando de um periférico, através da técnica de sondagem (pooling).

Exercício

O conversor A/D é usado para converter sinais analógicos (valores contínuos) para sinais no formato digital (valores discretos), para posterior tratamento pelo computador. O conversor usado no PG possui uma escala de -5V a +5V, correspondendo cada unidade do valor lido a cerca de 0,04V.

O conversor D/A é usado para converter valores numéricos num sinal analógico. O conversor D/A usado no PG converte um valor numérico entre 0 e 255 num sinal analógico entre -5V e +5V.

Usando o periférico PG juntamente com o Painel de Comando escreva um programa, em linguagem C, para realizar as seguintes operações:

- a) Digitalização de um som, captado pelo microfone.
- b) Reprodução do som, previamente digitalizado, no altifalante.

Nota 1: O ganho do pré-amplificador do microfone não é muito elevado. Para obter resultados visíveis (i. e., audíveis) deverá colocar o ganho máximo do pré-amplificador do microfone e falar perto deste quando estiver a usar o conversor A/D.

Nota 2: A velocidade de reprodução do som (i.é., o número de amostras por segundo) deve ser idêntica à da digitalização. Para implementar esta característica pode utilizar o conversor A/D para definir a cadência reprodução dos valores digitalizados.

Exercício 4: E/S através de Posições de Memória

Objectivos

- Familiarização com as portas de transferência de dados mapeadas em memória.
- Construção de programas para comando de um periférico, através da técnica de sondagem (pooling).

Exercício

O controlador gráfico do PC possibilita a geração de imagens em modo gráfico, onde o tom de cada pixel da imagem é controlado através de um endereço de memória. A linha dos computadores pessoais IBM PC compatíveis permite a geração de imagens em vários modos gráficos (CGA, EGA, VGA, XGA, etc.). Cada modo gráfico possibilita uma resolução (i.é., número de pixels) horizontal e vertical, bem como um número limitado de cores para cada pixel. Por exemplo, o modo VGA possui uma resolução de 640x480, onde cada pixel está associado a um bit (imagem a preto e branco) ou a 16 bits, o que permite a utilização de 64Kcores.

Embora a resolução e o número de bits utilizados para cada pixel possa ser definido directamente em registos da placa gráfica, tal é uma tarefa complexa, sendo preferível utilizar os modos gráficos pré-definidos. Estes modos gráficos podem ser programados através de chamadas a rotinas do sistema, acedidas por interrupções. Nomeadamente, a interrupção 10h permite seleccionar o modo gráfico. Neste exercício será utilizado o modo 13h, o qual possui uma resolução de 320x200 com 8 bits por pixel. O código que permite seleccionar este modo é o seguinte (o qual pode ser introduzido directamente num programa em C):

```
asm {  
    MOV AX, 0x13  
    INT 0x10  
}
```

Neste modo gráfico, cada célula de memória corresponde ao valor utilizado por um pixel. Em aplicações MS-DOS, o endereço da memória vídeo, utilizada pelo controlador para gerar a imagem, está acessível a partir do endereço 0xA0000. O acesso a estes endereços de E/S pode ser realizado através de um apontador, declarado em C da seguinte forma:

```
unsigned char *ptr = (unsigned char far *) 0xA0000000L;  
ptr [10] = 20; // atribui ao pixel (10,0) o tom 20
```

Usando o periférico PG juntamente com o Painel de Comando escreva um programa, em linguagem C, para ilustrar graficamente a forma de onda de um som, captado pelo microfone.

Exercício 5: E/S Controlado por Interrupções

Objectivos

- Construção de programas de E/S através da técnica de interrupção.

Exercício

Os sistemas de operação usam interrupções para controlar o teclado. Em MSDOS sempre que é premida uma tecla é accionada a interrupção 09h. Uma rotina, instalada especialmente para atendimento daquela interrupção, é usada para obter informação sobre as teclas premidas, com salvaguarda para uso posterior. Um conjunto adicional de funções são usadas para posteriormente analisar e tratar a informação recolhida.

Uma rotina de atendimento a uma interrupção pode ser instalada através do seguinte esqueleto:

```
void interrupt (* RotinaAntiga)(); // apontador para a rotina antiga

void interrupt RotinaNova() { // rotina a instalar
    ... // código da rotina nova
    (*RotinaAntiga)(); // chamada à rotina antiga
}

void main {
    RotinaAntiga = getvect(<<n°int>>); // endereço da rotina antiga
    disable(); // inibe interrupções
    setvect(<<n°int>>,RotinaNova); // instala a nova rotina
    enable();
    keep(0,0x1000); // termina o programa, ficando residente
}
```

- a) Escreva um programa, em linguagem C, usando a técnica de sondagem, que faça reflectir nos Leds a acção sobre as teclas direccionais (< e >). A função *kbhit()* permite identificar se foi premida uma tecla, enquanto a porta de E/S 0x60 contém o ASCII da última tecla premida. Considere que no conjunto de Leds do Painel de Comando se encontra sempre um e um só Led aceso. Considere no início o Led4 aceso. Pressionando na tecla “>” deve passar a acender o led à esquerda (direita) do actualmente aceso - até chegar ao extremo em que permanecerá o mesmo aceso.
- b) Adapte o programa anterior ao uso da técnica de interrupção.
- c) Adapte o programa do Exercício 3 para utilizar a técnica de interrupção. Note que a linha de interrupção IRQ 3 está associada ao vector de interrupção 0x0B