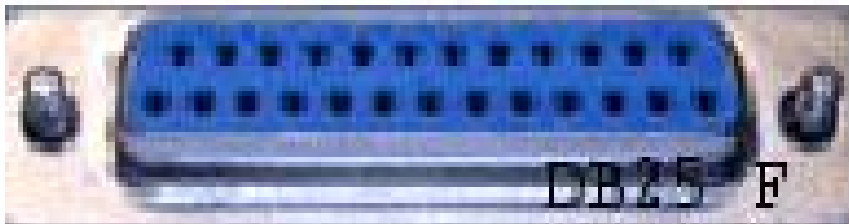


# Automação de protótipos

## Linguagem C e Porta Paralela

# Teoria de Porta Paralela

- » Porta Paralela - interface de comunicação com periféricos (impressoras, scanners, câmeras de vídeo, unidade de disco removível e outros);
- » Com algum conhecimento de programação e eletrônica, pode-se controlar um aparelho através da Porta Paralela do computador (conector DB25);



# Teoria de Porta Paralela

- » A porta paralela permite a alocação de 9 bits para entrada de dados e 12 bits para saída de dados.
- » Com eletrônica básica faz-se o controle de dispositivos externos ou a leituras de sensores.
- » Padronizada pela IEEE 1284 em 1994.



The background features a collage of four images: a whiteboard with the text 'I LOVE TRANSISTORS' and a stick figure, a person working with electronic components, a code editor window showing C++ code for a CD motor application, and a yellow printed circuit board (PCB) with various components.

# Teoria de Porta Paralela

» A porta paralela original possuía a capacidade para enviar bytes que representassem caracteres em ASCII para impressoras matriciais.



# Barramentos I/O

- » Envio ou recebimento de dados
  - Deve-se informar o endereço dos barramentos pelos quais a informação vai trafegar.
- » PORTA PARALELA:
  - » Endereço para o conjunto de 8 bits.
    - » A informação (entrada ou saída) trafega sempre em 8 bits!
  - » Valores utilizados pelo processador para identificar o caminho a ser utilizado:
    - » Dados: 0x378 (hex) | 888 (dec)
    - » Controle: 0x37A (hex) | 890 (dec)
    - » Status: 0x379 (hex) | 889 (dec)

# Barramentos de I/O

- » Modos de Operação:
  - » A porta paralela atualmente possui três modos de operações. São eles (IEEE 1284):
    - » SPP (Standard Parallel Port) – bits de dados unidirecional;
    - » EPP (Enhanced Parallel Port) – bits de dados bidirecional;
    - » ECP (Extended Capabilities Port) – bits de dados bidirecional;
  - » Estes modos de operação são configurados pelo BIOS Setup.
  - » Diferenciam-se:
    - » “Bidirecionabilidade”;
    - » Velocidade de transmissão;
    - » Forma de transmissão;
    - » Etc.

# Barramentos de I/O

» Pinagem da porta paralela:

STATUS Endereço 379h		CONTROLE Endereço 37Ah		DADOS Endereço 378h	
D7	11	D7	NC	D7	9
D6	10	D6	NC	D6	8
D5	12	D5	NC	D5	7
D4	13	D4	NC	D4	6
D3	15	D3	17	D3	5
D2	NC	D2	16	D2	4
D1	NC	D1	14	D1	3
D0	NC	D0	1	D0	2



# Padronização da Porta Paralela

## » Padronização IEEE 1284:

- Define uma comunicação serial de dados:
  - Informação enviada em fila (FIFO);
  - Porta paralela pode ser considerada como uma sendo comunicação BYTE serial.

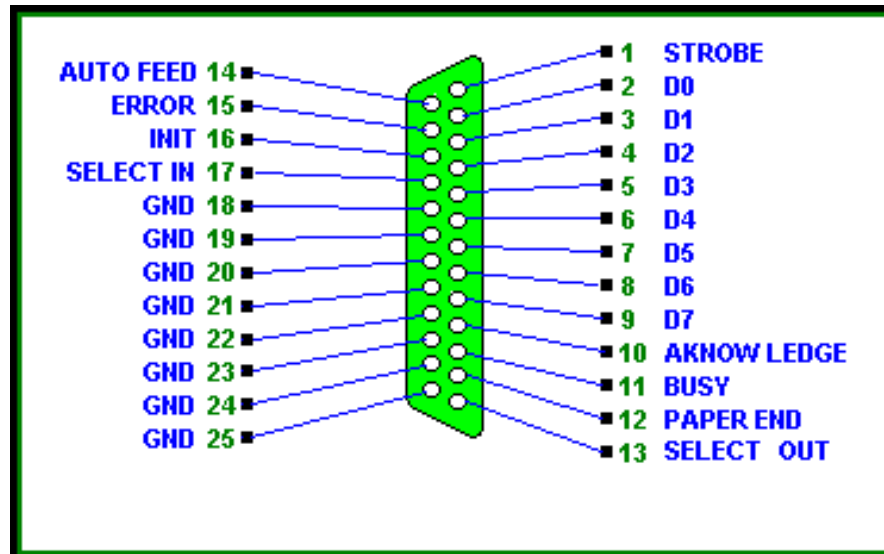
## » Padronização:

- Caracteres são enviados um a um como um conjunto de bits;
- Conectores DB-25;
- Recomendado para conexões curtas (<15 metros);
- Os sinais variam de 3 a 15 volts positivos ou negativos;
  - +-5, +-10, +- 12 e +-15 volts
  - O nível lógico um é definido por ser voltagem negativa e o nível lógico zero é voltagem positiva.

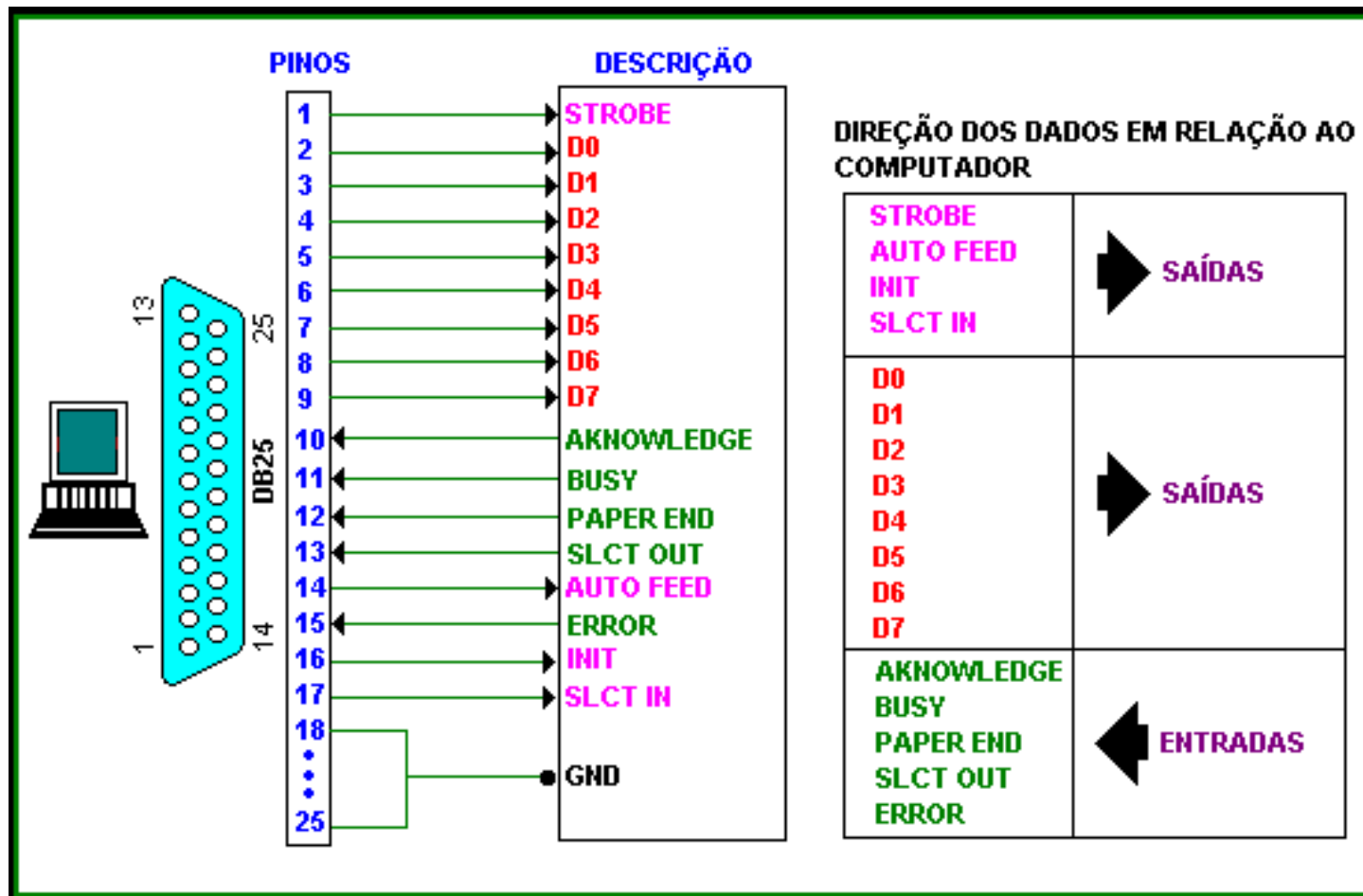
# Padronização da Porta Paralela

## » Padronização:

- Taxas de transmissão:
  - 300, 1200, 2400, 9600, 19200, etc (bits);
  - ambos os dispositivos devem estar configurados com a mesma velocidade;
- Pinagem:

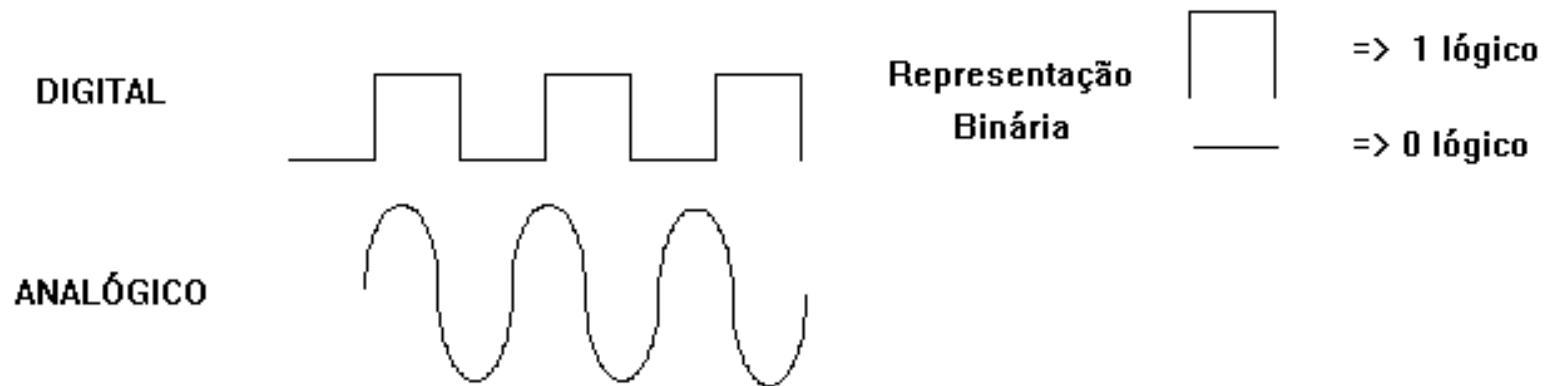


# Conector DB25 – Pinagem



# Formato do Sinal

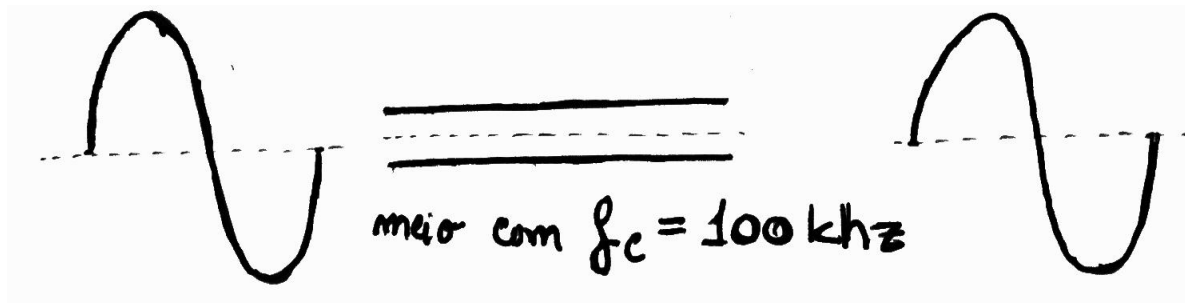
## » Forma dos sinais:



- » Sinal Analógico
  - » Frequência → ciclos/segundo [hz]
- » Sinal Digital
  - » Taxa de Transmissão → pulsos/segundo [bauds/s]
  - » Se representa dois níveis lógicos → [bits/s]

# Restrições à Passagem do Sinal

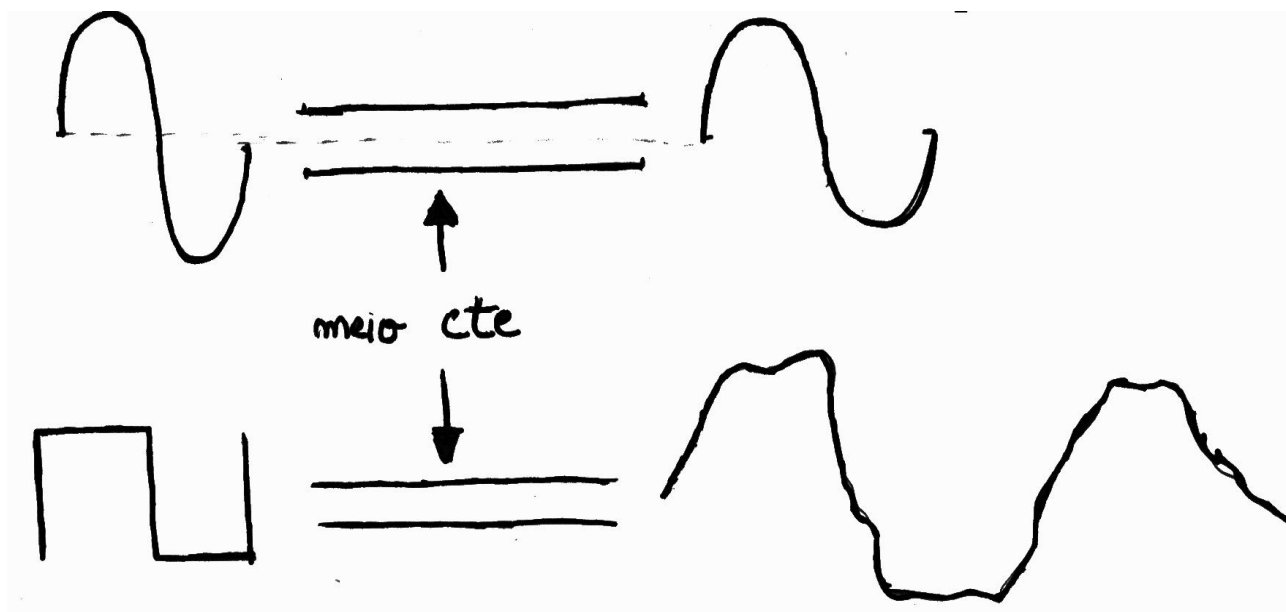
- » Banda Passante e Frequência de Corte (Filtro)
  - Todo meio de transmissão tem uma capacidade de passagem de “frequências” estabelecida pela frequência de corte
  - Assim um Sinal com  $F = 50 \text{ kHz}$ :



- » Sinal Analógico  $\rightarrow$  Harmônico Fundamental
- » Sinal Digital  $\rightarrow \infty$  Harmônicos
  - Fourier decompôs matematicamente.

# Restrição a passagem da informação

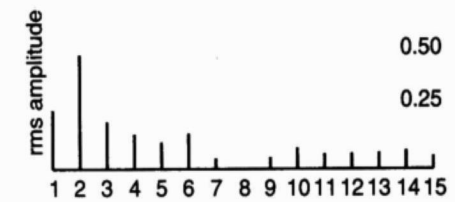
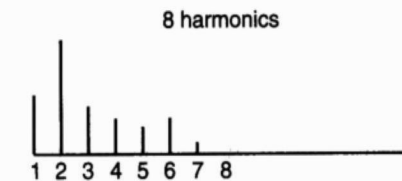
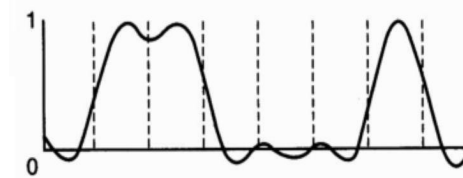
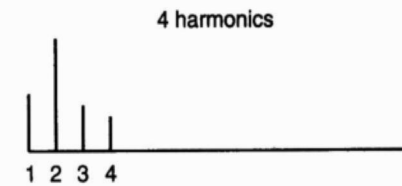
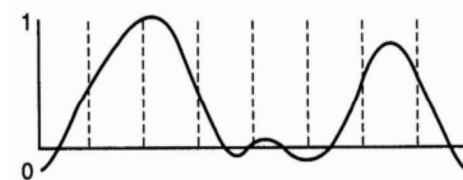
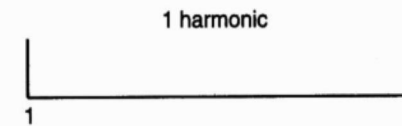
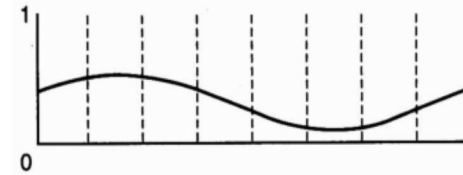
- » Sinal Analógico: passa/não passa
  - pequena degeneração
- » Sinal Digital tem perdas degenerativas



# Restrição a passagem da informação

» Analisando o Espectro do Sinal Digital:

Bps	T (msec)	First harmonic (Hz)	# Harmonics sent
300	26.67	37.5	80
600	13.33	75	40
1200	6.67	150	20
2400	3.33	300	10
4800	1.67	600	5
9600	0.83	1200	2
19200	0.42	2400	1
38400	0.21	4800	0

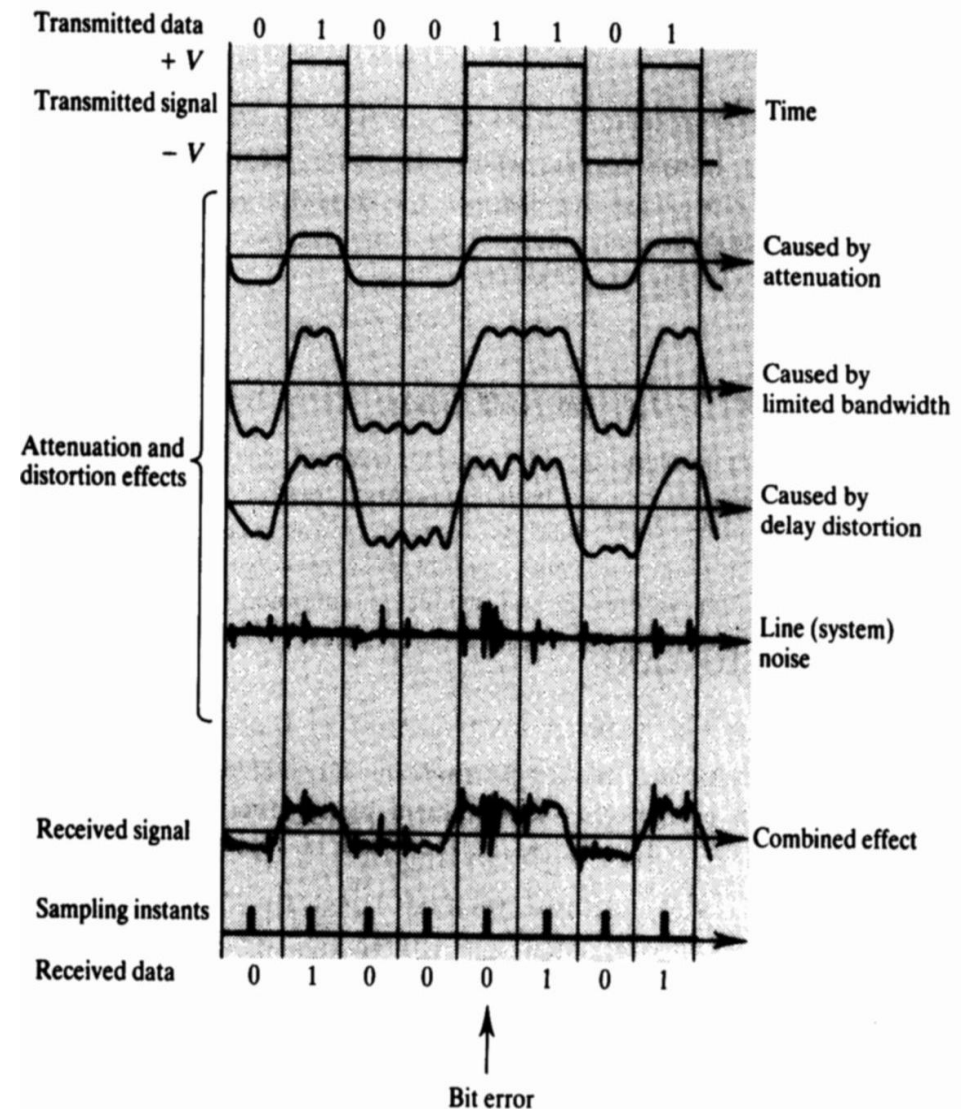


Time →

Harmonic number →

# Restrição a passagem da informação

- » **Conjunto das Restrições:**
- » Atenuação (Attenuation)
- » Filtro (Limited Bandwidth)
- » Ruído (Noise)
- » Atraso (Delay)
- » Comparativo dos Atrasos
  - Satélite: 257 ms
  - Microondas: 3  $\mu$ s/Km
  - Cabo Coaxial: 5  $\mu$ s/Km
- » Ruído Aleatório



# Acessando a porta paralela

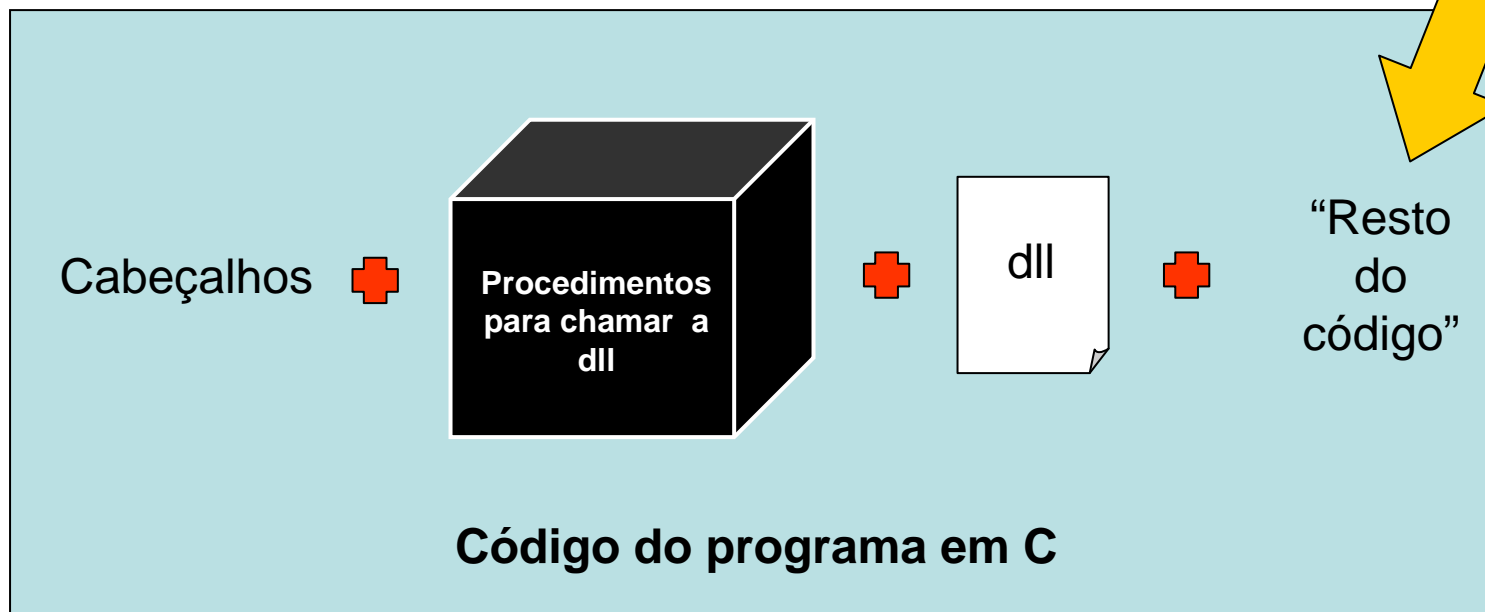
## » Sistema Operacional (“Maestro”);

- HAL (*Hardware Abstraction Layer* – camada de abstração de hardware): permite o “*plug and play*”
- Família Windows 9x , a maioria das linguagens de programação acessavam com facilidade a porta paralela através de funções nativas da própria linguagem ou via código assembler.
- Windows NT/2000/XP não permitem o acesso direto a este tipo de porta. Nesse caso é necessário um driver de sistema.



# Envio de Dados

- » Como o compilador identifica as função de envio e recebimento de dados pela porta paralela?
  - Utilizando funções para chamada de dll;
  - Chamando a dll dinamicamente;





# Acessando a porta paralela

## » Chamada a DLL

- Biblioteca será carregada dinamicamente durante a execução de um programa (quando se fizer necessária);
- Carregar certas funcionalidades conforme a demanda;
- Em alguns sistemas o programador precisa ser cauteloso para assegurar que a biblioteca seja carregada antes de ser chamada, enquanto outras também automatizam esse processo.

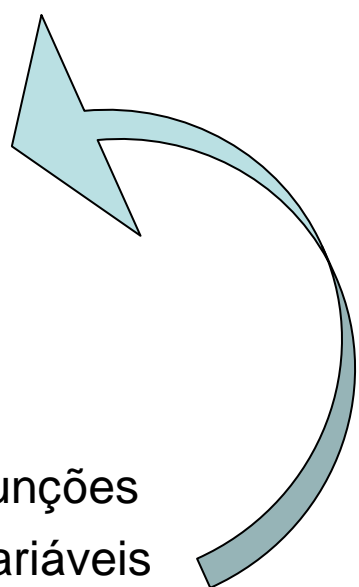
## » A Inpout32.dll possui duas funções: Out32 e Inp32.

- Out32 escreve um valor (byte) num endereço de I/O
- Inp32 lê um valor (byte) de um endereço de I/O.

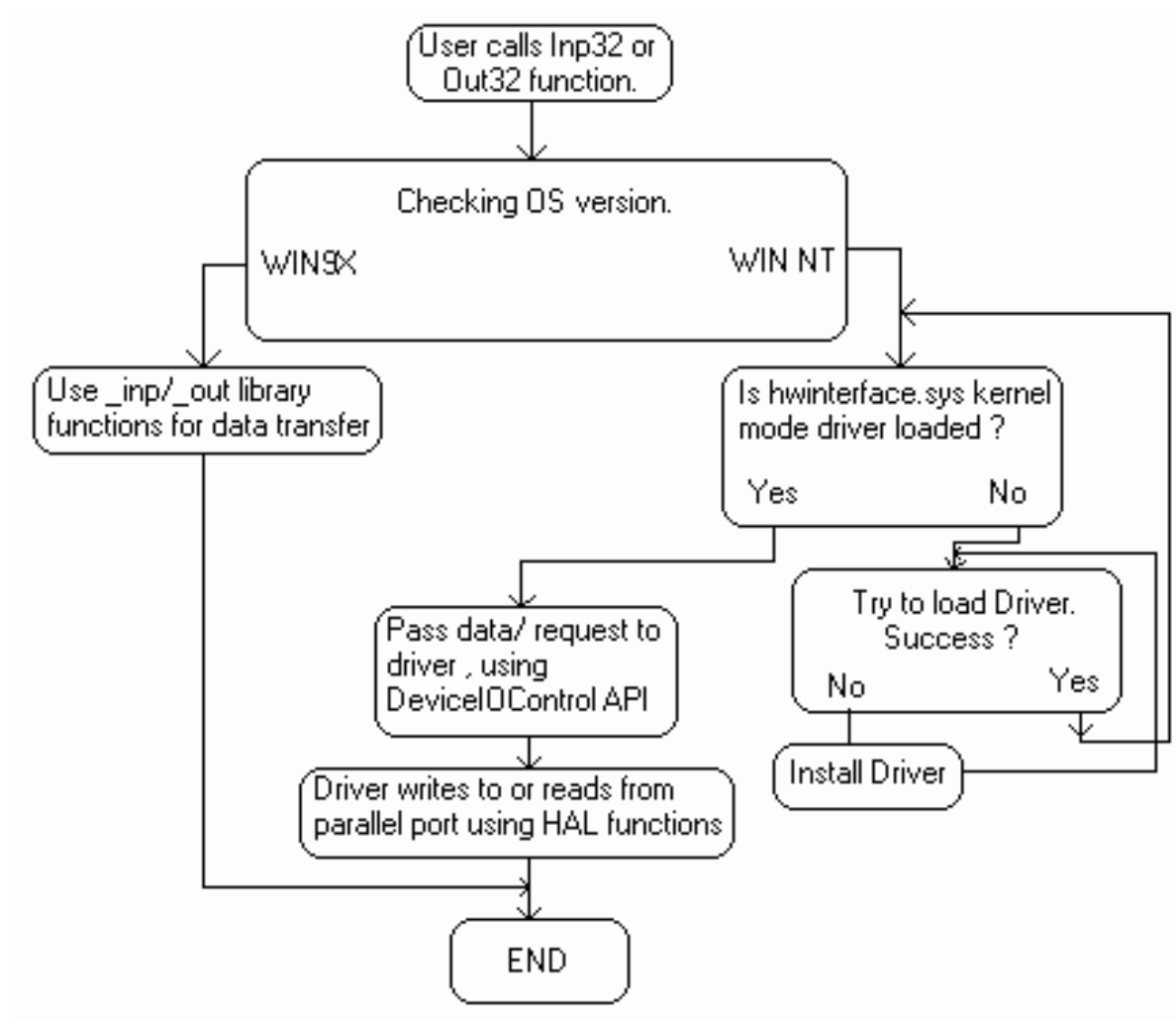
# Chamada a dll (“caixa preta”)

```
#include <conio.h>
#include <process.h>
#include <windows.h> //Necessário para: LoadLibrary(), GetProcAddress() e HINSTANCE.

//Declaração dos ponteiros para função.
typedef short _stdcall (*PtrInp)(short EndPorta);
typedef void _stdcall (*PtrOut)(short EndPorta, short valor);
HINSTANCE hLib; //Instância para a DLL inpout32.dll.
PtrInp inportB; //Instância para a função Imp32().
PtrOut outportB; //Instância para a função Out32().
int main(void)
{
    hLib = LoadLibrary("inpout32.dll"); //Carrega a DLL na memória.
    if(hLib == NULL) //Verifica se houve erro.
    {
        printf("Erro. O arquivo inpout32.dll não foi encontrado.\n");
        getch();
        return -1;
    }
    outportB = (PtrOut) GetProcAddress(hLib, "Out32"); //Obtém o endereço da função Out32 contida na DLL.
    if(outportB == NULL) //Verifica se houve erro.
    {
        printf("Erro. A função Out32 não foi encontrada.\n");
        getch();
        return -1;
    }
CÓDIGO!
    FreeLibrary(hLib); //Libera memória alocada pela DLL.
    return 0;
}
```

- 
- + Funções
  - + Variáveis
  - + Cabeçalhos
  - + implementações

# Acessando a porta paralela



# Envio de Dados

- » O comando em C que atribui valores referente à porta desejada é o: `outportB(ENDERECO, VALOR)`
- » Significado de tais parâmetros:
  - ENDERECO
    - São valores em hexadecimal que representa o endereçamento da porta desejada. Normalmente são:

Printer	Data Port	Status	Control
LPT1	0x378	0x379	0x37a
LPT2	0x278	0x279	0x37a
LPT3	-	-	-

# Conector DB25 – Pinagem

- » **Sobre os pinos de dados:** pinos de 2 à 9, normais, i.e., nível lógico alto (um) ativa. A letra D significa registrador de DADOS, e o número significa a posição do bit no byte.

Pinos de Dados								
Identificação	D7	D6	D5	D4	D3	D2	D1	D0
Pino	9	8	7	6	5	4	3	2
Posição	7	6	5	4	3	2	1	0
Byte	1	1	1	1	1	1	1	1

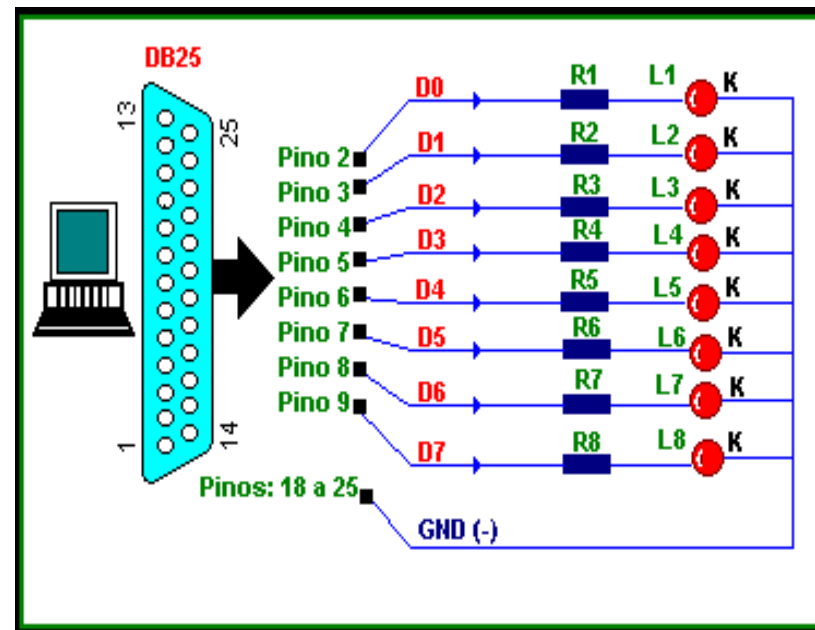
# Envio de Dados

- » `outportB(ENDERECO, VALOR);`
- » VALOR:
  - É o valor decimal da representação binária relacionada bit a bit com a pinagem da do conector DB25.

Decimal	Binário	Pino/Fio ativo (5V)	Comentário
1	00000001	2 - D0	Cada bit do byte enviado à Porta Paralela está relacionado com um pino do DB25, e um fio do cabo paralelo, fisicamente. Ao enviar um byte, que o(s) bit(s) esteja(m) ligado(s) ou desligado(s), os LEDs acende(rão) ou apaga(rão) conforme os estados dos bits.
2	00000010	3 - D1	
4	00000100	4 - D2	
8	00001000	5 - D3	
16	00010000	6 - D4	
32	00100000	7 - D5	
64	01000000	8 - D6	
128	10000000	9 - D7	

# Envio de Dados

- » Então, para ativar o pino 3 – D1 dos oitos pinos da porta de dados de endereço 0x378 através de comando em C:  
`outportB(0x378, 2);`



- » Estando o computador ligado ao circuito acima, este comando acenderia somente o LED L2.

# Conector DB25 – Pinagem

- » **Sobre os pinos de controle:** C0 (*Strob* – pino 1), C1 (*Auto Feed* – pino 14) e C3 (*Slct in* – pino 17) que são invertidos, i.e., nível lógico baixo (zero) ativa. Apenas o bit C2 (*Init* – pino 16) é normal, i.e., nível lógico alto (um) ativa. A letra C significa registrador de CONTROLE, e o número significa a posição do bit no byte.

Pinos de Controle								
Identificação	Nenhum				C3	C2	C1	C0
Pino	Nenhum				17	16	14	1
Posição	7	6	5	4	3	2	1	0
Byte	0	0	0	0	1	1	1	1

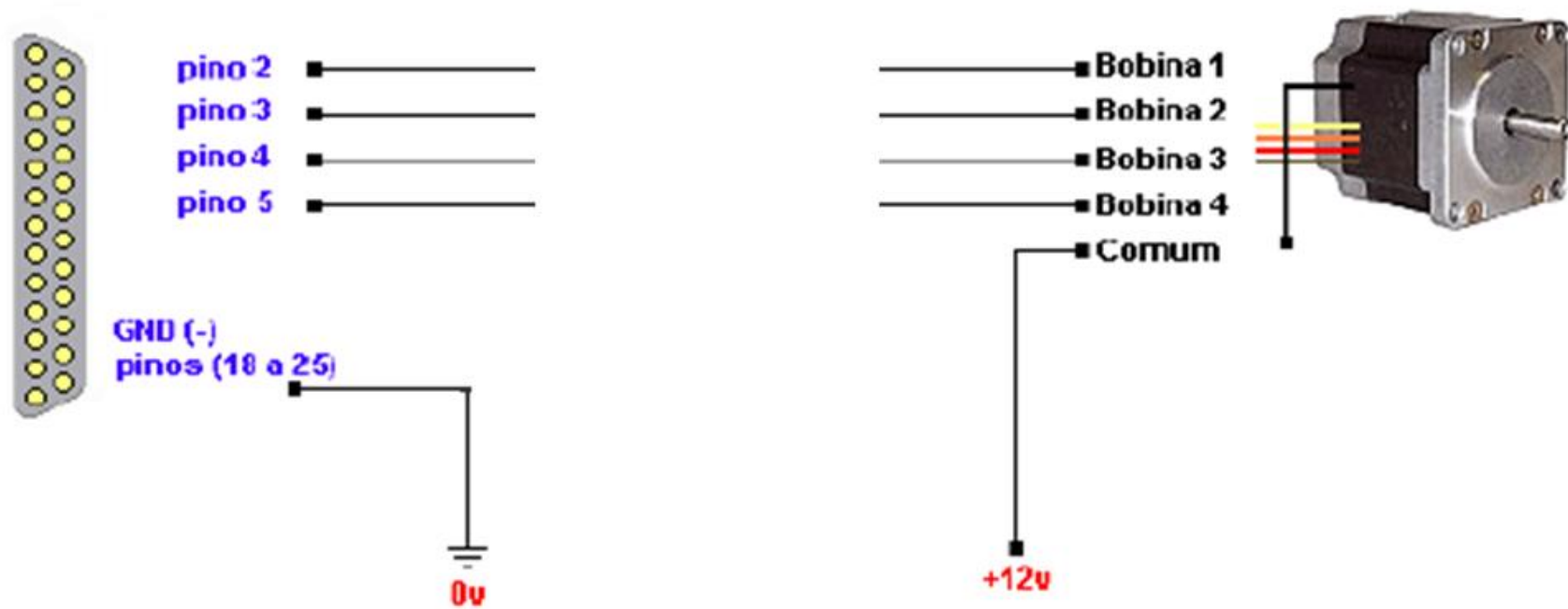
# Como acender os leds sequencialmente?

## » LOOP!

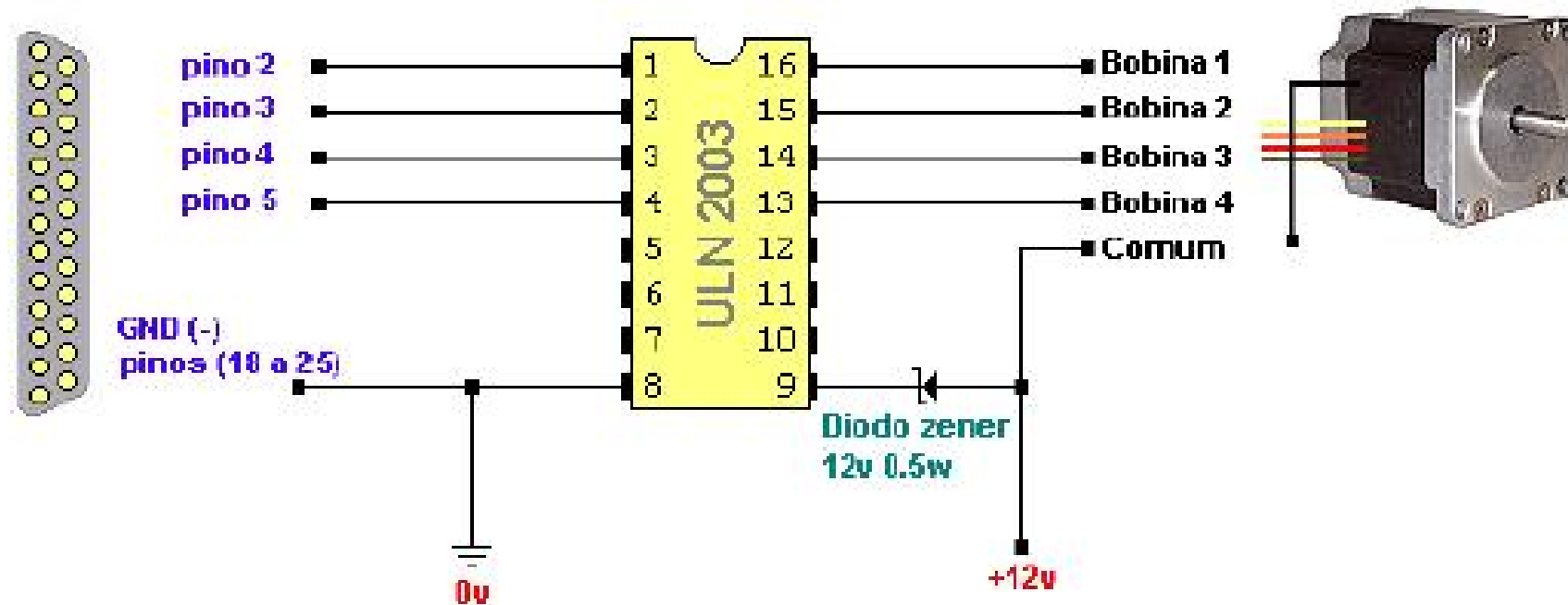
- for() / while() / do|while();
- Dependem da variável e da lógica utilizada;
- Exemplo:

```
#define LPT1 0x378
int i;
unsigned char enviar=128; //10000000 em binário.
for (i=0; i<7; i++)
{
    outportB(LPT1,enviar);           //Envia para a Porta LPT1 - Acende o
                                     LED colocado na posição enviar.
    enviar = enviar >> 1;           //A cada passagem, o bit 1 é movido
                                     para a direita.
    sleep(100);                       //Tempo para o próximo LED acender.
}
```

# Acionamento de motor de passo via porta paralela



# Acionamento de motor de passo via porta paralela



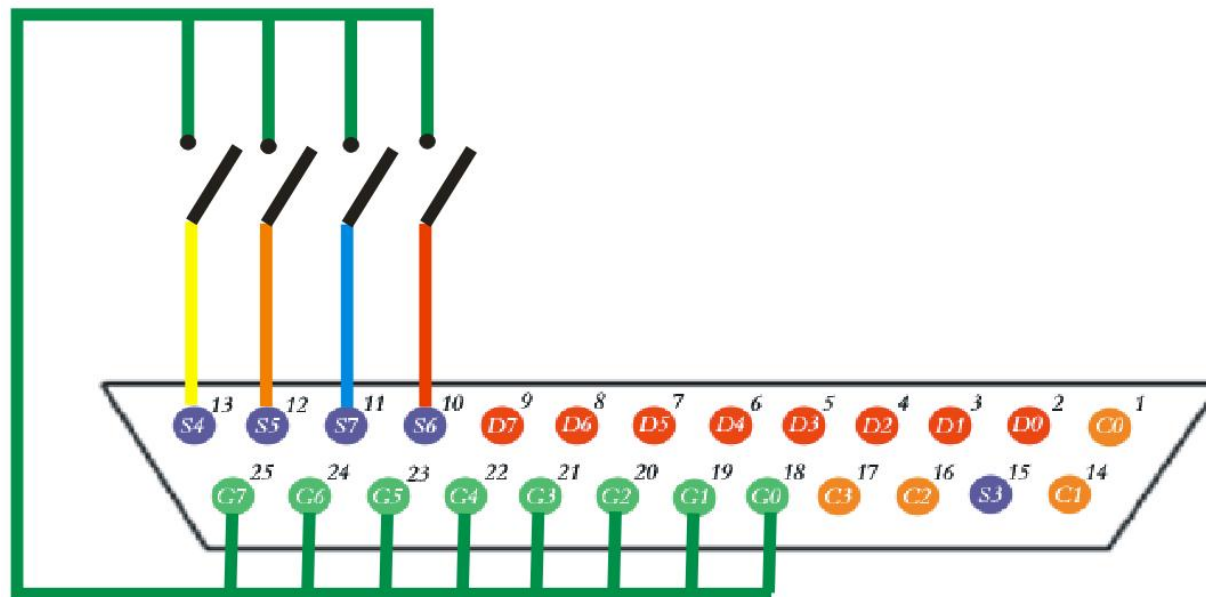
# Recebimento de dados

- » Variável = `inportB(ENDERECO)`
- » Valor retornado:
  - É o byte retornado pela porta paralela, com representação binária relacionada bit a bit com a pinagem da do conector DB25.
- » **Sobre os pinos de status:** S7 (*Busy* – pino 11) é invertido, i.e., nível lógico baixo (zero) ativa. Os outros bits, S6 (*Ack* – pino 10), S5 (*Paper end* – pino 12), S4 (*Slct out* – pino 13) e S3 (*Error* – pino 15) são normais, i.e., nível lógico alto (um) ativa. A letra S significa registrador de STATUS, e o número significa a posição do bit no byte.

Pinos de Status								
Identificação	S7	S6	S5	S4	S3	Nenhum		
Pino	11	10	12	13	15			
Posição	7	6	5	4	3	2	1	0
Byte	0	1	1	1	1	1	1	1

# Leitura de sensores via porta paralela

- » Pinos 18 a 25 GND da porta paralela
- » Pinos 10 a 13 entrada de dados da porta paralela
  - Ao fechar o curto da entrada com o terra o valor do pino é alterado



# Como receber o estado de um sensor?

» Controle de fluxo:

```
int main(void)
{
    int Byte;          // Armazena o byte recebido da porta
                      // paralela
    ...
    while ( ! kbhit() )    // executa enquanto nenhuma tecla for
                          // pressionada
    {
        Byte = inportB(0x379); // lê um byte da porta paralela
        printf(“%d\n”, Byte); // mostra na tela o valor
                              // correspondente em inteiro.
        ...
    }
}
```

# Cuidado !!!



- » A Porta Paralela está ligada diretamente à placa mãe de seu computador. Muito cuidado ao conectar circuitos eletrônicos a essa porta, pois, uma descarga elétrica ou um componente com a polaridade invertida, poderá causar danos irreparáveis ao seu computador, seja coerente.

