

Porta Paralela

Fabricio Ferrari
`www.ferrari.pro.br`

Universidade Federal do Pampa, Bagé, RS, Brazil

Maio 2008

Porta Paralela

- Porta de comunicação paralela
 - SPP Single Parallel Port 150 Kbits/s
 - EPP Enhanced Parallel Port 2 Mbits/s
 - ECP Enhanced Capabilities Port Canal DMA
- Palavra de 1 byte (8bits)
- Lógica TTL 0-5 volts
 - ~ 0.5V \longleftrightarrow 0, baixo, desligado
 - ~ 2.4V \longleftrightarrow 1, alto, ligado
- fornece 2.6 mA (saindo) ou 24 mA (entrando)
- Conector DB25, Centronics (impressoras)



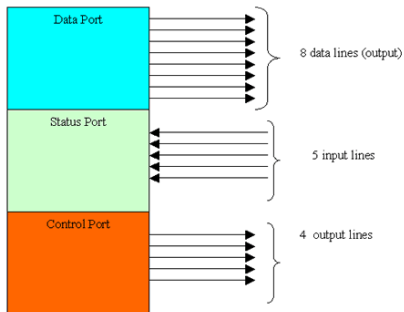
Endereçamento

ENDEREÇO **BASE** DE MEMÓRIA E/S:

0x278, 0x378, 0x3BC (usuais)

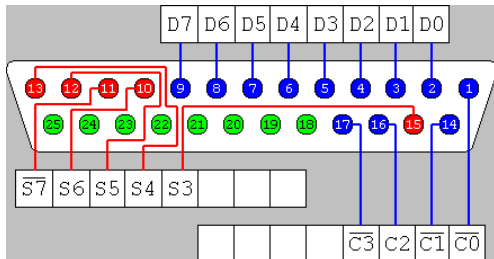
3 BYTES CONSECUTIVOS NA MEMÓRIA E/S

DATA	= BASE	BYTE 1 – escrita	0x278
STATUS	= BASE+1	BYTE 2 – leitura	0x279
CONTROL	= BASE+2	BYTE 3 – leitura/escrita	0x27a



Signal Name	Registrador Bit	DB-25 Pino	Direção E/S	Descrição
-Strobe	$\overline{C0}$	1	Output	sincronia: 0=escrita, 1=leitura
+Data Bit 0	D0	2	Output	dados
+Data Bit 1	D1	3	Output	dados
+Data Bit 2	D2	4	Output	dados
+Data Bit 3	D3	5	Output	dados
+Data Bit 4	D4	6	Output	dados
+Data Bit 5	D5	7	Output	dados
+Data Bit 6	D6	8	Output	dados
+Data Bit 7	D7	9	Output	dados
-Acknowledge	$\overline{S6}$	10	Input	
+Busy	$\overline{S7}$	11	Input	Ocupado
+Paper End	S5	12	Input	Sem papel
+Select In	S4	13	Input	
-Auto Feed	$\overline{C1}$	14	Output	Auto-alimentação
-Error	S3	15	Input	Erro
-Initialize	$\overline{C2}$	16	Output	Inicialização
-Select	$\overline{C3}$	17	Output	
Ground	-	18-25	-	Neutro

Registradores e Pinos



	7	6	5	4	3	2	1	0	I/O Port
DATA	x	x	x	x	x	x	x	x	Base = 278,
STATUS	\bar{x}	x	x	x	x	-	-	-	Base+1
CONTROL	-	-	-	-	\bar{x}	x	\bar{x}	\bar{x}	Base+2

Note: S7, C0, C1 & C3 are inverted

Habilitar permissões de acesso à porta.
superusuário/administrador (root)

```
#include <sys/io.h>
...
ioperm(BASE,BYTES,OPERACAO);
...
    BASE = 0x278
    BYTES = 3 (DATA, STATUS, CONTROL)
    OPERACAO = 1:habilita 0:desabilita

ioperm(0x278, 3, 1);
```

A partir do endereço (0x278),
(3) bytes consecutivos,
tem acesso habilitado (1).

Leitura

```
inb(ENDERECO);
```

Escrita

```
outb(VALOR, ENDERECO);
```

ENDEREÇO

0x278 (DATA), 0x279=0x278+1 (STATUS), 0x27a=0x278+2 (CONTROL)

ou

0x378 (DATA), 0x379=0x378+1 (STATUS), 0x37a=0x378+2 (CONTROL)

ou ...

Escrita

```
outb(0x8b, 0x278);
```

```
0x8B = 10001011b
```

Pino	valor
D7	1 5V
D6	0 0V
D5	0 0V
D4	0 0V
D3	1 5V
D2	0 0V
D1	1 5V
D0	1 5V

0x1 = 0001_b

0x2 = 0010_b

0x3 = 0011_b

0x4 = 0100_b

0x5 = 0101_b

0x6 = 0110_b

0x7 = 0111_b

0x8 = 1000_b

0x9 = 1001_b

0xA = 1010_b

0xB = 1011_b

0xC = 1100_b

0xD = 1101_b

0xE = 1110_b

0xF = 1111_b

Exemplo Básico de Escrita em C/Linux

Liga/desliga todos os pinos do byte DATA da porta paralela em 0x378

```
/* Exemplo de escrita na porta paralela
 * Liga e desliga os pinos DATA.
 * Compilar com:
 * gcc -O2 -o exemplo gravacao exemplo-gravacao.c
 * Fabrício Ferrari, 2008
 */

#include <stdio.h> // E/S
#include <sys/io.h> // ioperm(), outb(), inb()

int main(void) {

    // libera acesso aos 3 bytes da porta
    ioperm(0x378,3,1);

    // desliga todos os pinos
    outb(0x00, 0x378);

    // liga todos os pinos
    outb(0xFF, 0x378);

    // espera 1 segundos
    sleep(1);

    // desliga todos os pinos
    outb(0x00, 0x378);

    return(0);
}
```

Exemplo Completo de Escrita em C/Linux

Liga/desliga 10 vezes;

Define BASE, DATA, STATUS, CONTROL; verifica ioperm()

```
/* Exemplo de escrita na porta paralela
 * Liga e desliga os pinos DATA.
 * Compilar com:
 * gcc -O2 -o exemplo gravacao exemplo-gravacao.c
 * Fabrício Ferrari, 2008
 */
#include <stdio.h> // E/S
#include <stdlib.h> // Funcoes basicas
#include <sys/io.h> // ioperm(), outb(), inb()

#define BASE    0x378    // ou 0x278 ou 0x3BC
#define DATA   BASE
#define STATUS  BASE+1
#define CONTROLE BASE+2

int main(void) {
    int i;
    // libera acesso à porta e testa se deu certo
    if ( ioperm(BASE,3,1) == 1) {
        printf("Erro ao acessar porta no endereço 0x%X\n", BASE);
        exit(1);
    }

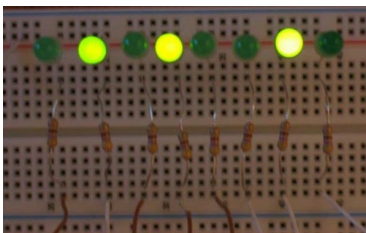
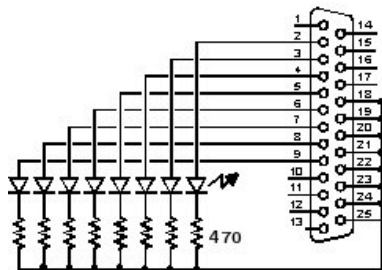
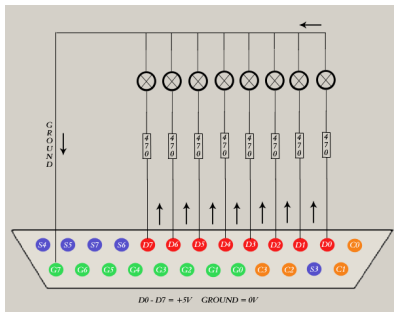
    for(i=0; i<10; i++){
        // liga todos os pinos de dados D0-D7 e espera 1s
        outb(0xFF, BASE);
        sleep(1);

        // desliga todos os pinos e espera 1s
        outb(0x00, BASE);
        sleep(1);
    }

    return(0);
} // Fim do Arquivo
```

Circuito de LEDs acionados pela porta

Porta fornecendo corrente



Ligando Pino a Pino DATA D0-D7

Define-se valores correspondentes a cada pino.
Seção do código em C correspondentes:

```
...
#define PIN01 0x01 // 1d 00000001b
#define PIN02 0x02 // 2d 00000010b
#define PIN03 0x04 // 4d 00000100b
#define PIN04 0x08 // 8d 00001000b
#define PIN05 0x10 // 16d 00010000b
#define PIN06 0x20 // 32d 00100000b
#define PIN07 0x40 // 64d 01000000b
#define PIN08 0x80 // 128d 10000000b

main(void){
    ...
    // liga so o pino 1
    outb(PIN01, BASE);
    // liga so o pino 2
    outb(PIN02, BASE);
    // liga so o pino 1
    outb(PIN03, BASE);
    ...
    // desliga todos
    outb(0x00, BASE);
    ...
}
```

Ligando Pinos Individuais

Operador OU-binário |

P.Ex., Ligando Pino 1 e Pino 5

```
outb( PIN01 | PIN05, BASE);
```

Operador OU-binário: |

	binário	hexadecimal
PIN01	00000001 _b	0x01
PIN05	00010000 _b	0x10
PIN01 PIN05	00010001 _b	0x11

Ligando Pino 1, Pino 2, Pino 3 e Pino 7

```
outb(PIN01|PIN02|PIN03|PIN07, BASE);
```

Leitura byte STATUS:

```
unsigned char valor; // variável de 1 byte 0-255
...
valor = inb(0x379); // retorna o valor do endereço 0x379
...                // BASE=0x378, STATUS=0x379, CONTROL=0x37A
```

Se nada conectado:

valor = 0x78 = 01111000

Bit	7	6	5	4	3	2	1	0
STATUS	S7	S6	S5	S4	S3	-	-	-
Pino	11	10	12	13	15	NC	NC	NC
Valor (0x78)	0	1	1	1	1	0	0	0

Lendo Pinos Individuais

Operador E-binário &

P.Ex., Lendo o valor do pino 10:

Leitura do byte STATUS:

```
valor=inb(STATUS)
```

Mascara para o bit 6 (pino 10)

```
(valor & 0x40)
```

Operador E-binário: &

	binário	hexadecimal
valor	01111000 _b	0x78
mascara	01000000 _b	0x40
(valor & mascara)	01000000 _b	0x40

Bit 6 ligado: $(\text{valor} \& \text{mascara}) = 0x40 = 01000000_b = 64_d$

Bit 6 desligado: $(\text{valor} \& \text{mascara}) = 0x00 = 00000000_b = 0_d$

Lendo Pinos Individuais – Cont.

Operador E-binário &

Deslocamento do bit – resultado 0 ou 1

Operador deslocamento-a-direita-binário: >>

valor >> deslocamento

	binário	hexadecimal
num	10000000 _b	0x80
num >> 1	01000000 _b	0x40
num >> 2	00100000 _b	0x20
num >> 3	00010000 _b	0x10
...

Voltando ao exemplo do pino 10:

pino10 = (valor & 0x40) >> 6

& 0x40 | separa o bit 6
>> 6 | desloca 6 casas p/direita

pino10 = 0 ou 1

STATUS

Pino	10	11	12	13	15
máscara	0x40	0x80	0x20	0x10	0x08
deslocamento	6	7	5	4	3

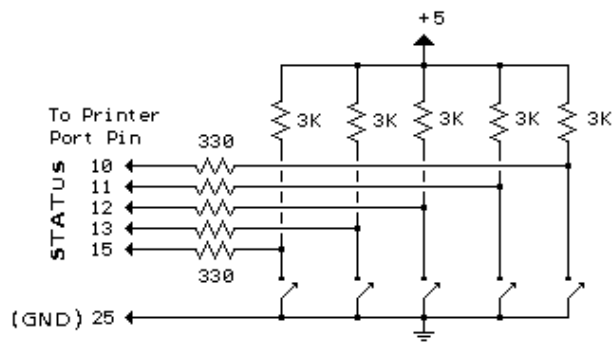
CONTROL

Pino	1	14	17
máscara	0x01	0x02	0x08
deslocamento	0	1	3

```
valorpino = (inb(ENDERECO) & mascara) >> deslocam ;
```

valorpino é 1 ou 0 (ligado ou desligado)

Circuito de Leitura



- Interfacing the Standard Parallel Port
<http://www.beyondlogic.org/spp/parallel.htm>
- Use of a PC Printer Port for Control and Data Acquisition
<http://et.nmsu.edu/etti/fall96/computer/printer/printer.html>
- Linux Parallel Port
<http://www.torque.net/linux-pp.html>
- Parallel Port Made Easy (Epanorama.net)
http://www.epanorama.net/circuits/parallel_output.html
- A Parallel Port Pin Programming Library for Linux
<http://parapin.sourceforge.net/>
- Controlling LEDs (Light Emitting Diodes) with Parallel Port
<http://www.codeproject.com/KB/cs/csppled.asp>
- Bowden's Hobby Circuits
http://ourworld.compuserve.com/homepages/Bill_Bowden/
- Linux I/O port programming mini-HOWTO
<http://tldp.org/HOWTO/I0-Port-Programming.html>

Fabrizio Ferrari, 2008
www.ferrari.pro.br