



Comércio e Manutenção de Produtos Eletrônicos
proxsys@proxsys.com.br

Nota de Aplicação KPP

Manual de utilização: Kit Porta Paralela

PROXSYS

Proposta

A proposta do kit de treinamento é servir como ferramenta de estudos e testes para aplicação da porta paralela do PC em projetos de eletrônica e automação, onde o número de entradas e saídas (I/O) não seja grande. Com pequenas implementações é possível controlar dispositivos externos através do seu computador.

O kit é excelente apoio nos estudos de eletrônica e programação para técnicos, engenharia elétrica, engenharia mecatrônica, engenharia da computação e outras.

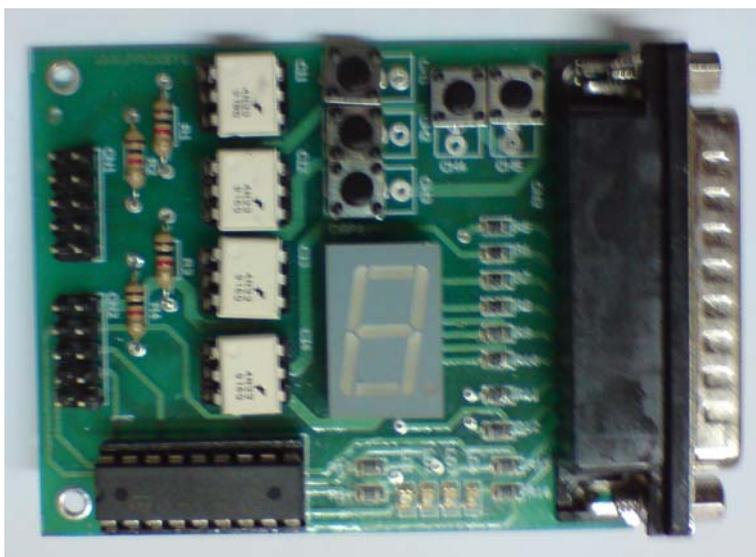
Uso do Kit Porta Paralela

São as mais variadas aplicações do kit, tais como:

- Controle para casas inteligentes
- Alarmes
- Automação de pequenos equipamentos
- Robótica
- Controle de iluminação
- Interface de sensores com o PC
- Estudo de programação e eletrônica

Com o aprendizado do funcionamento da porta paralela, aliado ao uso de uma boa ferramenta de programação e uma dose de criatividade, o emprego desta ferramenta não tem limites.

Características do Kit Porta Paralela



- 1 – Display de 7 segmentos catodo comum para criação de contadores, sinalizações, testes de saída e outras aplicações.
- 4 – Leds smd para testes de saída.
- 5 – chaves NA (normalmente aberta) para testes de entrada, simulação de sensores.
- 1 – Buffer de corrente ULN2803 para controle de dispositivos externos (Reles, motores de corrente continua, válvulas solenóides e outros) através do conector de expansão (CN2).
- 4 – Optoacopladores para a entrada de dados externos (Sensores, chaves, sinais digitais que podem ser outros controladores) através do conector de expansão (CN1).

Porta Paralela

É uma interface de comunicação entre o computador e um periférico externo. A história começa quando a IBM criou em 1980 seu primeiro PC (Personal Computer), a idéia inicial era conectar a porta paralela em uma impressora, mas atualmente, os dispositivos periféricos são os mais variados.

A figura 1 ilustra uma porta paralela:



Figura 1 – Porta paralela conectada ao PC

Conector DB-25

O conector DB-25 normalmente se localiza na parte de trás do computador. A figura 2 e a Tabela 1 ilustram a disposição dos pinos e suas funções.

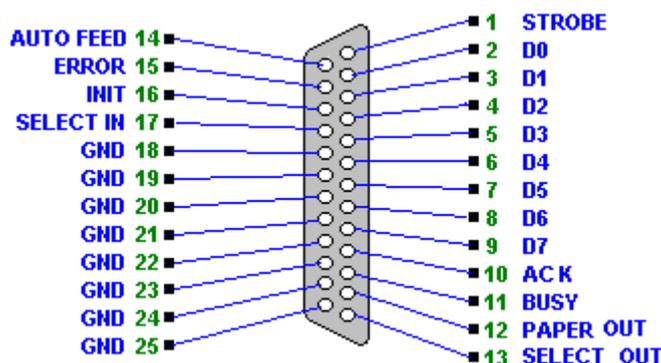


Figura 2 – Conector DB-25

PINO	FUNÇÃO	DIREÇÃO	TIPO	PINO	FUNÇÃO	DIREÇÃO	TIPO
1	Strobe	Saída	Controle	10	Ack	Entrada	Status
2	Dado 0 (D0)	Saída	Dados	11	Busy	Entrada	Status
3	Dado 1 (D1)	Saída	Dados	12	Paper Out	Entrada	Status
4	Dado 2 (D2)	Saída	Dados	13	Select Out	Entrada	Status
5	Dado 3 (D3)	Saída	Dados	14	Auto Feed	Saída	Controle
6	Dado 4 (D4)	Saída	Dados	15	Error	Entrada	Status
7	Dado 5 (D5)	Saída	Dados	16	Init	Saída	Controle
8	Dado 6 (D6)	Saída	Dados	17	Select In	Saída	Controle
9	Dado 7 (D7)	Saída	Dados	18-25	GND	Terra	

Tabela 1 – Descrição dos pinos da porta paralela

Acesso à porta Paralela

A porta paralela pode ser acessada tanto em nível de byte quanto de bit. Mas não podemos acessar os pinos de entrada e saída diretamente. O Acesso é feito através de registradores de controle da porta. A tabela 2 demonstra os endereços dos registradores de Dados, Controle e Status:

Nome	Endereço (LPT1)	Endereço (LPT2)	Descrição
Registro de Dados	378h	278h	Envia 1 byte para a impressora(saída de dados)
Registro de Status	379h	279h	Lê o status da impressora (entrada de dados)
Registro de Controle	37Ah	27Ah	Envia dados de controle para a impressora (saída de dados)

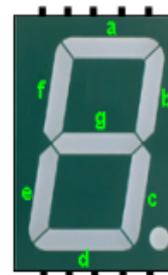
Tabela 2 – Endereço dos registradores

Acessando o Registro de Dados (Acionando o Display de 7 segmentos)

O registrador de dados trabalha com lógica positiva, ou seja, ao receber “1” lógico em um de seus bits, o pino é ligado passando para nível lógico “1” (+5 Vdc), o mesmo acontecendo para “0” lógico quando então o pino é levado a 0 Vdc (GND).

A tabela 3 demonstra a disposição dos pinos com seus valores respectivos:

Pino	Descrição	Valor em Hexadecimal a ser inserido	Resultado do resgistrador	Disposição do Display 7 segmentos kit
2	D0	01h	00000001b	Segmento A
3	D1	02h	00000010b	Segmento B
4	D2	04h	00000100b	Segmento C
5	D3	08h	00001000b	Segmento D
6	D4	10h	00010000b	Segmento E
7	D5	20h	00100000b	Segmento F
8	D6	40h	01000000b	Segmento G
9	D7	80h	10000000b	Segmento H



Display 7 Segmentos

Tabela 3 – Valores para o endereço de dados

Exemplo: Deseja-se formar o algarismo 1 do display de 7 segmentos, para isso é necessário que os segmentos B (D1) e C (D2) estejam ligados:

Delphi:

```
Valor := $2 + $4; //carrega os valores correspondentes aos segmentos B e C.
```

```
Outportb ($378, Valor); //envia o valor para o registrador de dados.
```

Visual Basic:

```
Valor = &H2 + &H4 'carrega os valores correspondentes aos segmentos B e C.
```

```
Outportb &H378, Valor 'envia o valor para o registrador de dados.
```

Acessando o Registro de Controle (Acionando os Leds)

Para trabalhar com os pinos de controle, é preciso conhecer a posição dos mesmos no registrador de controle 37Ah (LPT1), conforme a tabela 4 abaixo:

Pino na porta paralela					17 (Select In)	16 (Init)	14 (Feed)	1 (Strobe)	
Posição do pino no registrador de Controle	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	
Valor lido	X	X	X	X	1	0	1	1	
					LD4	LD3	LD1	LD2	Leds Kit

Tabela 4 – Posição dos pinos no registro de controle e seus estados

As posições marcadas com “X” mostram que não há pinos de entrada/saída conectados a esses bits. Um detalhe importante sobre os pinos de controle diz respeito a sua lógica de ativação. Nos pinos de dados, o nível lógico de ativação é igual a “1” e desativação igual a “0”. Porém para os pinos de controle, exceto o pino 16, operam com lógica invertida. Ou seja, para que os pinos da porta 1,14 e 17 sejam levados a +Vcc é necessário inserir nível lógico “0” no registrador.

Exemplo: Deseja-se acionar o LD3 e LD4:

Para isso é necessário colocar nível lógico “0” no bit3 e nível lógico “1” no bit2.

Bit 6	Bit 5	Bit 4	Bit 3	LD4 Bit 3	LD3 Bit 2	LD1 Bit 1	LD2 Bit 0
0	0	0	0	0	1	1	1
Como esses bits tanto faz o valor, então pode-se carregar com "0"				LD4 e LD3 ligado		LD1 e LD2 desligado	

Tabela 5 – Ligando o LD3 e LD4

O valor a ser enviado para o registro de controle é 7h.

Delphi:

Valor := \$7; //carrega os valores correspondentes aos LD3 e LD4.

Outportb (\$37A, Valor); //envia o valor para o registrador de dados.

Visual Basic:

Valor = &H7 'carrega os valores correspondentes aos LD3e LD4.

Outportb &H37A, Valor 'envia o valor para o registrador de dados.

Acessando o Registro de Status (Chaves – Sensores)

Temos apenas cinco pinos de entrada a serem lidos na porta, para a leitura correta é necessário saber a posição de cada pino, no byte do registrador de status (379h para LPT1). Veja a tabela 6 abaixo:

	Chave Kit					NC	NC	NC
	CH4	CH2	CH1	CH5	CH3			
Pino na porta paralela	11 (Busy)	10 (Ack)	12 (Paper Out)	13 (Select Out)	15 (Auto Feed)			
Posição do pino no registrador Status	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Valor lido	0	1	1	1	1	1	1	0

Tabela 6 – Disposição das chaves na placa

Os pinos usam a parte mais significativa do byte (bit 4 a bit 7) e um bit na parte menos significativa (bit 3). Os bits 2,1 e 0 não estão conectados fisicamente a placa, mas estão presentes na leitura do registrador.

Exemplo: Deseja-se verificar a CH4 e escrever uma mensagem na tela (“A chave 4 foi acionada”):

Para isso vamos isolar o nosso bit fazendo uma mascara e saber se ele esta acionado ou não.

Delphi:

```
Valor := inportb($379); //função de leitura do registro de status
```

```
If ( (Valor and $80) = $80 ) Then //realiza a mascara
showmessage('A CHAVE 4 FOI ACIONADA'); //escreve a mensagem na tela
```

Para o exemplo acima foi isolado o bit 7, do byte inserido no endereço 379h fazendo um AND lógico com o valor 80h (10000000b). Se o estado lógico do bit desejado for “1” (chave pressionada, por exemplo), o valor retornado será 80h (validando o If na comparação de igual) e escrevendo a mensagem na tela, se igual a “0” (chave não pressionada, por exemplo), o valor retornado será 0h não validando a comparação.

	CH4	CH2	CH1	CH5	CH3	NC	NC	NC
Valor Porta	0	1	1	1	1	1	1	0
Não acionada	0	1	1	1	1	1	1	0
Operação AND	1	0	0	0	0	0	0	0
Resultado	0	0	0	0	0	0	0	0

Tabela 7 – Operação AND com a CH4 não acionada

	CH4	CH2	CH1	CH5	CH3	NC	NC	NC
Valor da Porta chave Accionada	1	1	1	1	1	1	1	0
Operação AND	1	0	0	0	0	0	0	0
Resultado	1	0	0	0	0	0	0	0

Tabela 8 – Operação AND com a CH4acionada

Acessando dispositivos externos (Entradas e Saídas)

A tabela 9 mostra as configurações das expansões através dos conectores CN1 (entradas externas) e CN2 (saídas externas).

DB-25		CN1	
12	PAPER OUT	2	GND_FOTOS
13	SELEC OUT	4	FOTO 1 (CH1)
10	ACK	6	FOTO 2 (CH5)
15	ERROR	8	FOTO 3 (CH2)
		10	FOTO 4 (CH3)
DB-25		CN2	
18 / 25	GND	1	VCC
16	INIT	2	GND
17	SELECT IN	3	RELE 1
1	STROBE	4	RELE 2
14	FEED	5	RELE 3
4	D2	6	RELE 4
5	D3	7	RELE 5
2	D0	8	RELE 6
3	D1	9	RELE 7
		10	RELE 8

CN1

1	2
3	4
5	6
7	8
9	10

STR2X5

CN2

1	2
3	4
5	6
7	8
9	10

STR2X5

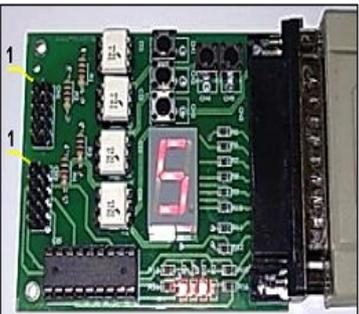


Tabela 9 – Entradas e saídas externas

Acesso à saída externa

O acesso aos registros das saídas é realizado nos exemplos anteriores. A figura 3 demonstra um exemplo de ligação externa para o CN2

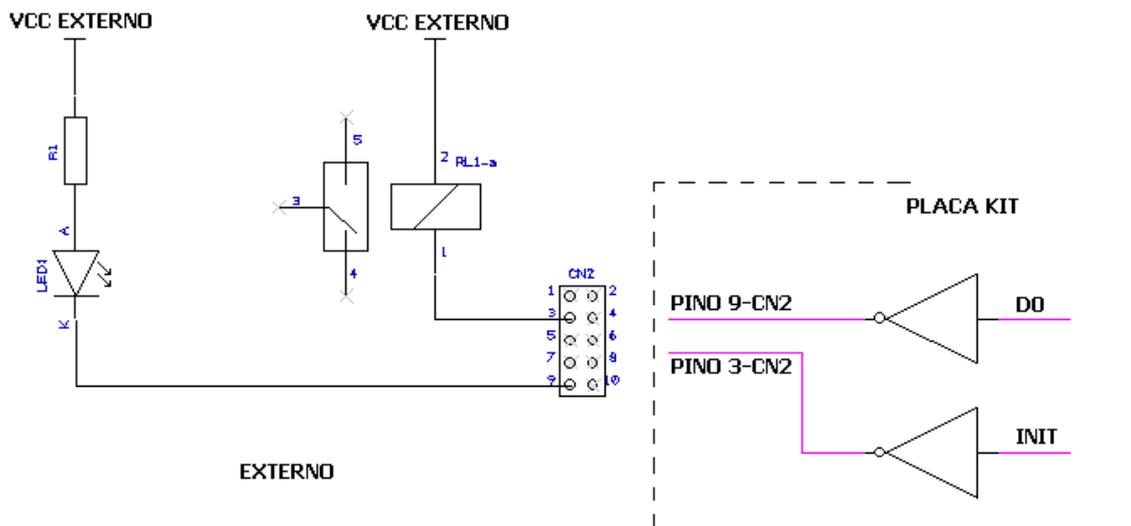


Figura 3 – Acionando o rele e um LED

Acessando à entrada externa

O acesso aos registros de entrada é realizado nos exemplos anteriores. A figura 4 demonstra um exemplo de ligação externa para o CN1

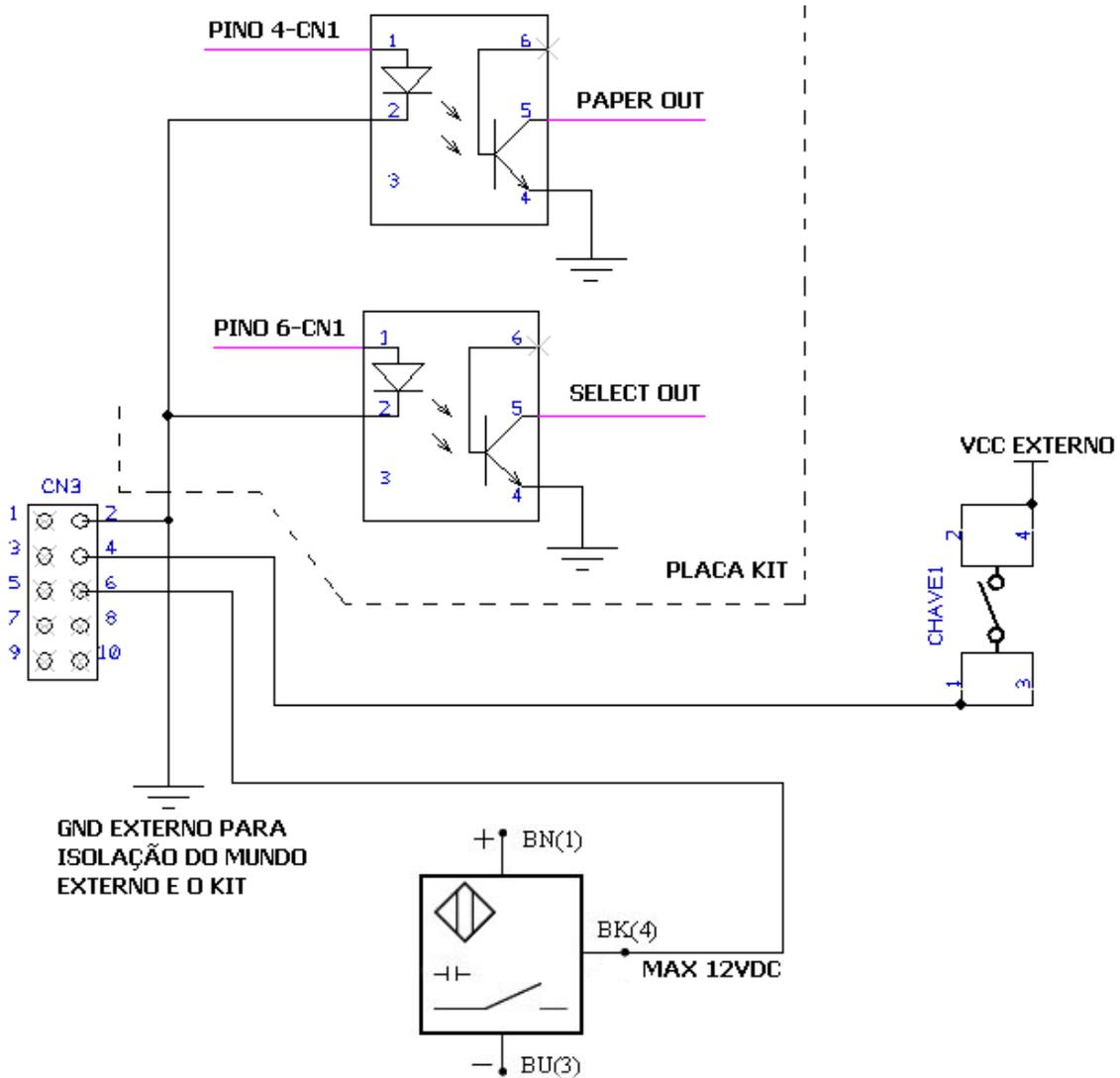


Figura 4 – Conectando sensores ao CN1

Para tensões de entrada superiores a 12 Vdc substituir os resistores respectivos (R1, R2, R3 e R4).

Acessando a porta paralela através do Visual Basic

Para acessar a porta paralela, é necessário colocar o arquivo **InpOut32.dll** (dll responsável por acessar a porta paralela do windows NT e XP) que esta salvo no CD do exemplo do kit (**Exemplo Visual Basic\InpOut32**) no diretório “C:\WINDOWS\system” do seu computador e adicionar o modulo **Porta.bas** que esta localizado em “**Exemplo Visual Basic\Modulo Acesso**” no projeto do Visual Basic.

Acessando a porta paralela através do Delphi

Para acessar a porta paralela, é necessário abrir o arquivo UserPort que esta localizado em “**Exemplo Delphi\UserPort**” e seguir as instruções no documento chamado **Acesso.doc**.