

# Aquisição de Dados Usando Logo e a Porta de Jogos do PC

C. E. Aguiar e F. Laudares

*Instituto de Física, Universidade Federal do Rio de Janeiro*

*Cx.P. 68528, Rio de Janeiro, 21945-970, RJ*

Recebido em 10 de Abril de 2001. Aceito em 3 de Janeiro 2002.

Descrevemos um sistema de aquisição de dados baseado na porta de jogos do IBM-PC com plataforma Windows. Os programas de aquisição e análise são totalmente escritos na linguagem Logo. O sistema é utilizado para medir o período de um pêndulo como função da amplitude de oscilação.

A data acquisition system using the game port of an IBM-PC running Windows is described. The programs for data acquisition and analysis are all written in Logo. The system is used to measure the period of a pendulum as function of the oscillation amplitude.

## I Introdução

O computador é um ótimo instrumento de laboratório. Ele presta-se muito bem a experiências que envolvem medidas de tempo, aquisição de dados em grandes quantidades, e tratamento de dados em tempo real [1]. Apesar de todas estas características favoráveis, os computadores têm sido pouco usados nos laboratórios didáticos. E quando isto ocorre, é quase sempre com o auxílio de “kits” pré-fabricados - pacotes de circuitos eletrônicos e programas produzidos por empresas especializadas, vendidos a preços relativamente altos, e que muitas vezes operam como verdadeiras caixas-pretas. Em geral é difícil usar estes pacotes para realizar experimentos diferentes daqueles para os quais eles foram projetados, o que limita a sua utilidade didática [2]. Esta falta de flexibilidade tem origem, em parte, nos programas de aquisição e tratamento de dados contidos nos kits, que raramente podem ser modificados ou mesmo compreendidos por professores e estudantes.

Neste trabalho descrevemos como montar um sistema de aquisição de dados simples, de baixo custo, utilizável em laboratórios didáticos da escola média, e que dá a alunos e professores controle completo sobre o seu uso. O sistema está baseado na substituição do “joystick”, em geral usado para controlar jogos no computador, por sensores que podem ser utilizados em uma grande variedade de experiências. A idéia de usar a interface de jogos para aquisição de dados não é nova. Entretanto, propostas anteriores [3, 4, 5] utilizavam computadores que não existem mais, como o Apple II e

MSX, ou programas de aquisição escritos em linguagens pouco usadas hoje, como Pascal e Basic (que deram origem ao Delphi e VisualBasic, muito populares mas pouco apropriados ao ensino médio). Uma implementação recente desta idéia pode ser encontrada na referência [6], com programas de aquisição escritos em C e QBasic. No que se segue descreveremos como montar um sistema de aquisição de dados a partir da porta de jogos de um IBM-PC com plataforma Windows (95/98/ME), usando programas totalmente escritos em Logo. Esta é uma linguagem simples, desenvolvida para uso didático, e que freqüentemente é ensinada na escola fundamental e média. Com isto todos os aspectos do sistema que descrevemos podem ser compreendidos e modificados por estudantes e professores, proporcionando-lhes uma grande flexibilidade no planejamento e execução de experimentos. A versão do Logo que utilizamos é o SuperLogo, produzido pelo Núcleo de Informática Aplicada à Educação (NIED) da Unicamp [7]. Esta versão tem os comandos em português, e é distribuída gratuitamente via Internet. Uma proposta semelhante à deste trabalho, usando o MSWLogo (em inglês) [8], pode ser encontrada na referência [9].

O motivo principal para se usar a porta de jogos em um laboratório didático é a segurança e facilidade com que podemos conectar sensores a esta interface. Um joystick é essencialmente um conjunto de resistências variáveis cujos valores são lidos a cada momento pelo computador. Substituindo o joystick por um componente eletrônico cuja resistência dependa da grandeza física que queremos medir (temperatura ou intensi-

dade luminosa, por exemplo) podemos monitorar esta grandeza, registrar a sua evolução temporal em intervalos muito pequenos, e manipular quantidades de dados que dificilmente seriam alcançáveis em um laboratório didático convencional.

Este artigo está organizado da seguinte maneira. Na seção II discutimos o funcionamento do joystick e da porta de jogos [10, 11]. Uma forma simples de medir tempos é apresentada na seção III. Na seção IV descrevemos sucintamente alguns sensores que podem ser conectados à porta de jogos. Na seção V mostramos um experimento realizado com o sistema. Alguns comentários gerais e conclusões estão na seção VI.

## II A porta de jogos

### II.1 O joystick

O tipo mais comum de joystick para IBM-PC são os joysticks analógicos. Estes joysticks têm dois botões e dois potenciômetros cujas resistências vão de 0 a 100 k $\Omega$  (em alguns casos até 150 k $\Omega$ ). O movimento esquerda-direita do joystick (eixo X) muda a resistência de um dos potenciômetros, e o movimento frente-trás (eixo Y) muda a resistência do outro. Em geral as resistências são nulas quando o joystick está todo para a esquerda e para a frente. Um esquema do joystick está mostrado na Fig. 1. Os dois botões estão representados pelas chaves A e B, e X e Y são os potenciômetros já mencionados. A tensão de 5 V e o aterramento não são fornecidos pelo joystick, mas pela porta de jogos à qual ele é conectado.

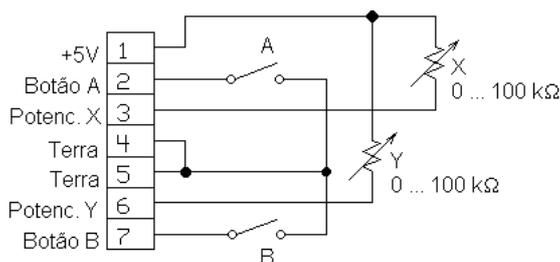


Figura 1. Esquema de um joystick

### II.2 A porta de jogos

A porta de jogos do IBM-PC é uma interface para dois joysticks analógicos, embora exista a conexão para apenas um deles. O uso simultâneo de dois joysticks só é possível com um cabo especial em forma de 'Y'. O joystick é conectado à parte traseira do computador via um soquete de 15 pinos, como o mostrado na Fig. 2.

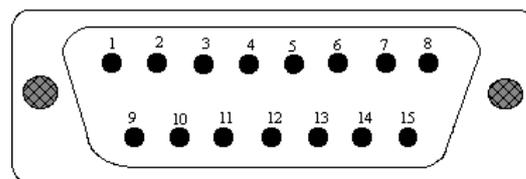


Figura 2. O soquete onde é conectado o joystick.

A porta de jogos não faz parte da "placa mãe". Em geral ela é implementada em uma placa própria ou, mais freqüentemente, colocada na placa de som. Neste último caso o conector tem dois dos seus 15 pinos dedicados à porta MIDI (*musical instruments digital interface*). Os outros pinos dão acesso aos quatro botões e quatro potenciômetros (dois eixos X e dois Y) correspondentes aos dois joysticks, a um potencial de +5 Volts, e a um terra. Esta porta é muito segura, mas devemos tomar cuidado com curto-circuitos entre o potencial de 5 V e o terra.

A função dos pinos da porta de jogos pode ser vista na tabela 1. Cada um dos joysticks (que chamamos 1 e 2) tem dois botões (A e B) e dois potenciômetros (X e Y). Assim A1 é o botão A do joystick 1, X2 é o potenciômetro X do joystick 2, etc. Muitos joysticks dão acesso aos 4 botões. Nestes casos A1 e B1 referem-se aos botões principais, e A2 e B2 aos secundários.

PINO	FUNÇÃO
1	+5 Volts
2	Botão A1
3	Potenciômetro X1
4	Terra
5	Terra
6	Potenciômetro Y1
7	Botão B1
8	+5 Volts (ou sem uso)
9	+5 Volts
10	Botão A2
11	Potenciômetro X2
12	Terra (ou porta midi)
13	Potenciômetro Y2
14	Botão B2
15	+5 Volts (ou porta midi)

Tabela 1. Pinos da interface de jogos

### II.3 Como a porta de jogos trabalha

A Fig. 3 mostra os circuitos ligados a cada um dos 15 pinos da porta de jogos. Os circuitos conectam os botões e potenciômetros dos joysticks a 8 linhas b0...b7. Estas linhas alimentam 8 bits (1 byte), numerados respectivamente de 0 a 7, colocados no endereço de entrada/saída 513 (decimal). Os bits de 4 a 7 informam

o estado dos botões de forma bastante direta. Como vemos na Fig.3, quando um botão é pressionado o potencial na linha de saída torna-se 0 V, o que coloca o bit correspondente no valor 0. Quando o botão é solto o potencial da linha sobe para 5 V e o valor do bit passa a ser 1.

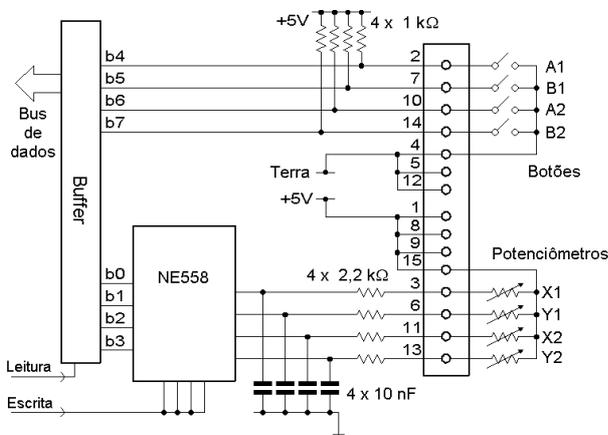


Figura 3. Esquema da porta de jogos.

Os bits de 0 a 3, correspondentes às linhas b0...b3, são usados para determinar a resistência dos potenciômetros. Esta determinação é mais complicada que no caso dos botões, e é feita com o auxílio de um circuito integrado, o NE555, composto basicamente por 4 “multivibradores monoestáveis” [12]. O circuito mantém os bits 0...3 com valor zero até que se escreva algo no endereço da porta. Neste instante circuito descarrega os capacitores ligados ao potenciômetros e coloca os bits 0...3 no estado 1. Cada capacitor começa então a ser carregado através do seu potenciômetro, e o bit correspondente é mantido no valor 1 até que a tensão no capacitor atinja cerca de 3,3 V (2/3 do valor máximo 5 V). A partir daí o circuito coloca o bit de volta no estado 0. O tempo  $T$  durante o qual o bit permanece no estado 1 depende da resistência  $R$  do potenciômetro segundo a fórmula

$$T(\mu s) = 24,2 + 11 \times R(k\Omega)$$

Encontrando este tempo obtemos a resistência do potenciômetro. A medida de tempo pode ser feita com um programa simples de contagem [5, 9]. Tal programa deve ser muito rápido, e portanto não pode ser escrito em uma linguagem interpretada como Logo. Por isto o SuperLogo tem um comando especial para determinar o tempo  $T$ , que será discutido mais à frente. Observe que com  $R = 100 \text{ k}\Omega$  gastamos aproximadamente 1 ms em uma medida de resistência. Note também que se não houver nada conectado a uma determinada linha ( $R = \infty$ ) o bit correspondente pode ficar indefinidamente com valor 1.

A tabela 2 resume como o estado dos joysticks é mapeado no byte da porta. A numeração dos bits, de 0 a 7, corresponde à significância deles no byte (os de menor índice estão “à direita” dos de maior índice).

BIT	JOYSTICK	VALOR
7	Botão B2	0=fechado, 1=aberto
6	Botão A2	0=fechado, 1=aberto
5	Botão B1	0=fechado, 1=aberto
4	Botão A1	0=fechado, 1=aberto
3	Potenciôm. Y2	1=medindo, 0=inerte
2	Potenciôm. X2	1=medindo, 0=inerte
1	Potenciôm. Y1	1=medindo, 0=inerte
0	Potenciôm. X1	1=medindo, 0=inerte

Tabela 2. Bits da porta de jogos.

## II.4 Lendo a porta de jogos

Muitas versões do Logo têm comandos que lêem a porta de jogos. No SuperLogo isto é feito com `portaentradab 513`, onde 513 é o endereço da porta de jogos (512 também serve). Se tivermos um joystick conectado à porta e executarmos a instrução

```
mostre portaentradab 513
```

obteremos o byte da porta, ou melhor, a sua representação decimal (um número decimal entre 0 e 255). Para obter o estado de cada uma das 8 linhas da porta (o valor de cada bit) devemos converter este número para a representação binária. A maneira mais eficiente de fazer isto é usar a operação lógica  $E$ , definida por:  $0E0 = 0, 0E1 = 0, 1E0 = 0, 1E1 = 1$ . Existe no SuperLogo uma operação, `bite`, que toma dois números inteiros como argumento e aplica a operação  $E$  a cada par de bits de mesma significância (mesma “posição” no byte). Por exemplo, aplicando `bite` aos números binários 1100 e 1010 obtemos 1000. Como o Logo só usa a representação decimal, esta operação é realizada com a instrução

```
mostre bite 12 10
```

já que  $12_{dec} = 1100_{bin}$  e  $10_{dec} = 1010_{bin}$ . A resposta que o Logo apresenta é 8, pois  $8_{dec} = 1000_{bin}$ .

Então, se quisermos saber qual é o  $N$ -ésimo bit ( $N = 0, 1, 2 \dots$ ) de um número  $X$ , basta fazer o `bite` de  $2^N$  com  $X$ . Todos os bits de  $2^N$  valem 0, com exceção do  $N$ -ésimo bit que vale 1. Portanto, se o  $N$ -ésimo bit de  $X$  for 0 o resultado da operação será 0. Se o  $N$ -ésimo bit for 1 o resultado será  $2^N$ . Por exemplo, para saber se o botão A1 (que corresponde ao bit 4) está apertado ou não, tomamos o `bite` de  $2^4 = 16$  com a leitura da porta:

```
mostre (bite 16 portaentradab 513)/16
```

Se o botão estiver apertado o resultado será 0, e se ele estiver livre o resultado será 1. Para saber o valor de cada um dos 8 bits da porta, basta aplicar o procedimento acima para  $N = 0, 1, 2, 3 \dots 7$ , o que corresponde a usar 1, 2, 4, 8...128 em bite. Isto pode ser feito com a instrução

```
mostre mapeie[(bite ? portaentradab 513)/?]
[128 64 32 16 8 4 2 1]
```

que produz uma lista com o valor dos oito bits da porta.

Com estes procedimentos podemos usar os botões do joystick para controlar programas Logo. E o que é mais importante, podemos monitorar sensores externos. Por exemplo, fotodiodos e fototransistores são componentes eletrônicos que conduzem corrente dependendo da incidência de luz (veja a seção IV). Substituindo um botão do joystick por algum destes fotosensores, podemos usar o computador para monitorar se o sensor está sendo iluminado ou não.

## II.5 Medindo resistências com a porta de jogos

Como já comentamos, a resistência de um potenciômetro do joystick é obtida medindo-se o tempo que o bit correspondente permanece com valor 1. Este tempo é tipicamente menor que alguns milissegundos. Fazer um programa em Logo para medi-lo não é eficiente, pois tal programa seria muito lento. Para realizar esta medida o SuperLogo tem uma função especial, `portajoystick :M`. Esta fornece o tempo que um bit da porta de jogos, especificado pela variável M, permanece no estado 1. Os valores  $M = 1, 2, 4, 8$  correspondem aos bits 0, 1, 2, 3, ou seja, aos potenciômetros X1, Y1, X2, Y2. Por exemplo, para medir a resistência de X1 executamos a instrução

```
mostre portajoystick 1
```

Devemos ter algum cuidado ao tentar medir os potenciômetros X2 e Y2. Como eles quase nunca estão ligados à porta (pois o segundo joystick raramente é usado) a resistência que é encontrada é infinita, e o tempo a ser medido também. Neste caso o comando `portajoystick` fornece o valor -1, mas pode levar muito tempo (até alguns minutos, dependendo do computador) para fazer isto.

O tempo fornecido por `portajoystick` não está em unidades predeterminadas. Mas, como já vimos, este tempo varia linearmente com a resistência, o que é suficiente para muitas aplicações. De qualquer forma, usando resistências conhecidas é possível calibrar a saída de `portajoystick`. Com este comando podemos utilizar o “manche” do joystick para controlar progra-

mas Logo. E, principalmente, podemos monitorar a resistência de sensores externos como fotoresistores e termistores (resistores dependentes de luz e temperatura, veja a seção IV) colocados no lugar dos potenciômetros.

## III A medida de tempo

Agora que já sabemos ler a porta de jogos, o próximo passo é registrar a evolução temporal dos sinais que ela recebe. Para isto precisamos aprender a medir o instante em que o sinal foi lido. No SuperLogo isto pode ser feito com o comando `tempomili`, que fornece o tempo em milissegundos decorrido desde que o sistema operacional (Windows) foi iniciado. Executando a instrução

```
mostre tempomili
```

obtemos a quantos milissegundos o Windows está operando. É importante saber que o tempo fornecido por `tempomili` não é atualizado a cada milissegundo, mas em intervalos cuja magnitude depende do computador e da configuração do sistema. Resoluções temporais típicas de `tempomili` estão na faixa de 3 a 15 ms.

Quando lemos apenas o estado dos botões, é possível obter uma resolução temporal melhor do que a citada acima. Quer o botão esteja aberto ou fechado, uma medida do seu estado leva sempre o mesmo tempo para ser realizada. Como este tempo é bem menor que a resolução temporal de `tempomili`, podemos usar a contagem de medidas como relógio. O intervalo de tempo entre duas medidas sucessivas é calculado dividindo o tempo total de aquisição, que pode ser medido com `tempomili`, pelo número de medidas realizadas. Este procedimento não pode ser utilizado se lemos um potenciômetro (via `portajoystick`) pois, como já vimos, neste caso a duração da medida depende do valor da resistência.

## IV Conectando sensores à porta de jogos

No lugar dos botões e potenciômetros do joystick, podemos conectar à porta de jogos componentes eletrônicos de resistência variável e usá-los como sensores. Nesta seção discutiremos rapidamente alguns desses componentes: o termistor NTC, o fotoresistor, o fotodiodo e o fototransistor [12, 13].

### IV.1 Termistores

O termistor é um componente eletrônico cuja resistência varia consideravelmente com a temperatura. Em geral os termistores são do tipo NTC (*negative temperature coefficient*), para os quais a resistência diminui

com o aumento da temperatura. Um termistor NTC é feito de material semiconductor, e é tipicamente utilizado na faixa de temperaturas entre  $-50\text{ }^\circ\text{C}$  e  $150\text{ }^\circ\text{C}$ .

A relação entre a resistência e a temperatura absoluta de um NTC (curva característica  $R/T$ ) é altamente não-linear, e pode ser aproximada por  $R = \alpha \exp(\beta/T)$ . Medindo a resistência para duas temperaturas diferentes podemos obter as constantes  $\alpha$  e  $\beta$  e construir uma curva de calibração razoavelmente precisa. A resistência de um NTC é geralmente especificada a  $25\text{ }^\circ\text{C}$ , e na maioria dos casos está entre  $100\ \Omega$  e  $100\ \text{k}\Omega$ . Tipicamente, a sensibilidade a  $25\text{ }^\circ\text{C}$  é da ordem de  $-4\% / ^\circ\text{C}$ .

### IV.2 Fotoresistores

O fotoresistor, também chamado de célula fotocondutora, fotocélula, ou LDR (*light dependent resistor*) possui uma resistência que depende da quantidade de luz que ele recebe. No escuro os fotoresistores têm resistência elevada, tipicamente na faixa de  $\text{M}\Omega$ . Quando são expostos à luz, a resistência diminui por ordens de magnitude. Sua desvantagem como sensor está na lentidão de resposta, que chega a décimos de segundo - uma fotocélula pode não perceber o “pisca-pisca” de 60 Hz de uma lâmpada fluorescente.

### IV.3 Fotodiodos

O fotodiodo é um diodo semiconductor em que a junção está exposta à luz. A corrente reversa de um fotodiodo aumenta linearmente com a incidência de luz, como pode ser visto nas curvas características  $I/V$  da Fig. 4. No escuro a corrente reversa é muito pequena, geralmente da ordem de nA. Observe que as curvas características passam pelo quarto quadrante ( $V \times I < 0$ ), de modo que o fotodiodo pode produzir energia elétrica - as células solares são um tipo de fotodiodo.

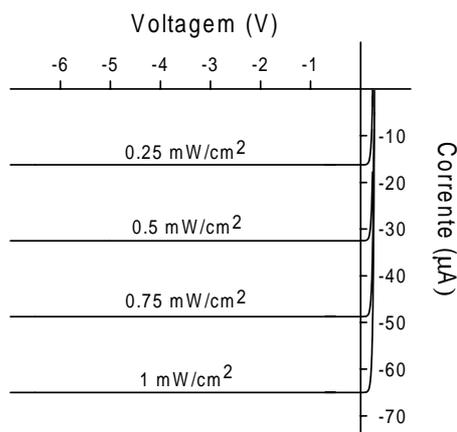


Figura 4. Curvas características  $I/V$  de um fotodiodo, para diferentes intensidades luminosas. Os valores mostrados são apenas representativos; dependendo do tipo, fotodiodos têm sensibilidades diferentes. Note que o fotodiodo opera com polarização reversa.

Os fotodiodos são úteis como sensores de radiação visível e infravermelha. A resposta espectral (sensibilidade a diferentes comprimentos de onda) de um fotodiodo de silício atinge o máximo em torno 900 nm, no infravermelho. Alguns fotodiodos têm coberturas que filtram a luz visível ou a radiação infravermelha.

As correntes obtidas com um fotodiodo são geralmente baixas, menores que  $100\ \mu\text{A}$  para uma incidência luminosa de  $1\ \text{mW}/\text{cm}^2$ . Com isto não podemos usar um fotodiodo diretamente no lugar de um botão do joystick. Como vemos na Fig.3, com o botão apertado passam pelo circuito cerca de 5 mA (na verdade  $\sim 4\ \text{mA}$  já são interpretados como “botão apertado” pela porta). Para obter correntes desta magnitude temos que amplificar o sinal do fotodiodo, ligando-o a um transistor como está mostrado na Fig.5.

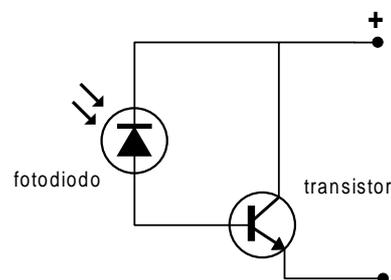


Figura 5. Circuito para amplificar a corrente de um fotodiodo.

### IV.4 Fototransistores

Os fototransistores são transistores com a junção coletor-base exposta à luz. Fototransistores são bem mais sensíveis que fotodiodos, gerando correntes da ordem de mA quando iluminados a  $1\ \text{mW}/\text{cm}^2$ . Um fototransistor com “ganho” suficientemente alto pode ser usado diretamente no lugar de um botão do joystick. Caso contrário um circuito semelhante ao da Fig.5 (um *fotodarlington*) pode ser usado para amplificar a corrente.

## V Aplicação: Período do pêndulo a grandes oscilações

### V.1 A montagem do experimento

O objetivo da experiência é medir o período de um pêndulo como função da amplitude de oscilação. O aparato utilizado está mostrado na Fig.6. O movimento do pêndulo é monitorado a partir das interrupções que a sua passagem causa no feixe luminoso que vai da lâmpada ao fotosensor.

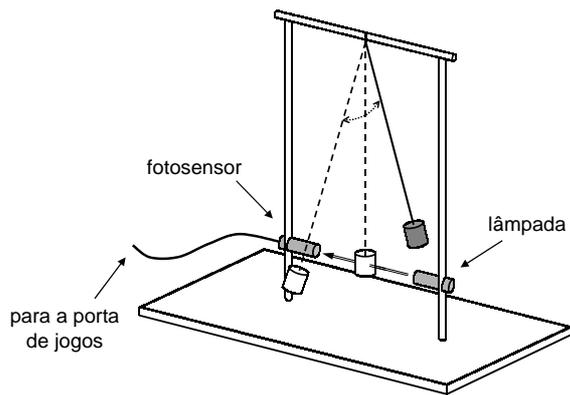


Figura 6. Esquema do arranjo experimental.

Usamos como detetor o conjunto fotodiodo-transistor discutido na seção IV.3, iluminado por um LED infravermelho obtido em um controle remoto de TV. O fotodiodo foi tirado de um *mouse* sem uso, e a amplificação foi feita por um transistor NPN de uso genérico (BC548C). A Fig. 7 mostra o sistema usado. O transistor está ligado à porta de jogos via os pinos 2 e 4 do conector, que correspondem ao botão A1 do joystick e ao terra (veja a seção II.2). Assim, quando o fotodiodo estiver no escuro, ou recebendo uma iluminação insuficiente, o transistor se comportará como uma chave aberta. Se o fotodiodo receber um sinal luminoso mais forte o transistor permitirá a passagem de uma corrente alta entre o pino 2 e o terra, simulando o apertar do botão do joystick. Portanto, quando o pêndulo passa entre o LED e o fotodiodo, deixando este último na sombra, o bit 4 da porta vale 1. Quando a luz atinge o fotodiodo o bit 4 passa a valer 0.

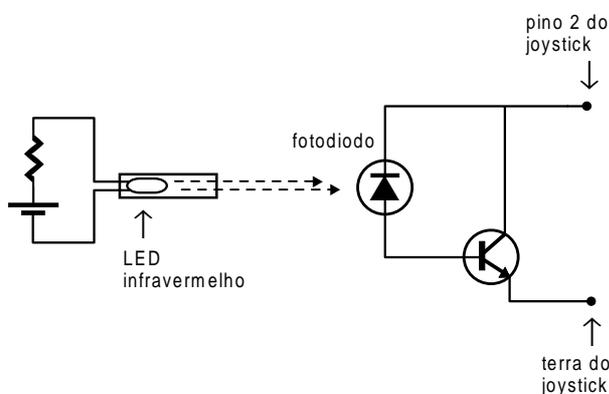


Figura 7. Diagrama do sistema de detecção. A “lâmpada” é um LED infravermelho e um fotodiodo é usado como sensor.

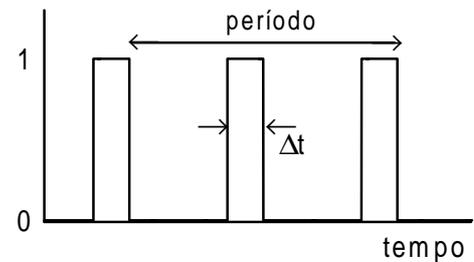


Figura 8. Sinal no detetor gerado pela oscilação do pêndulo.

Ao oscilar, o pêndulo gera então um sinal como o mostrado na Fig.8. Registrando este sinal podemos obter o período da oscilação, e o tempo  $\Delta t$  durante o qual o pêndulo esteve em frente ao fotosensor. A velocidade do pêndulo neste ponto é dada por

$$V_0 = \frac{D}{\Delta t},$$

em que  $D$  é o diâmetro do objeto oscilante ou, mais exatamente, a distância que este objeto percorre enquanto a porta recebe “1” como sinal. Esta distância pode ser medida independentemente, o que permite calcular  $V_0$ . Como o fotosensor está em frente à posição de equilíbrio do pêndulo,  $V_0$  é a velocidade máxima da oscilação. A velocidade angular correspondente é

$$\Omega_0 = \frac{V_0}{R} = \frac{D/R}{\Delta t},$$

em que  $R$  é a distância do feixe luminoso ao ponto de fixação do pêndulo. Na nossa montagem  $D = 3,3 \pm 0,1$  cm e  $R = 48,8 \pm 0,5$  cm. Conhecendo  $\Omega_0$  podemos calcular o ângulo máximo de oscilação,  $\Theta_{max}$ , dado por

$$\cos \Theta_{max} = 1 - \frac{R_G}{2g} \Omega_0^2$$

onde  $R_G$  é o raio de giração do pêndulo e  $g$  é a aceleração da gravidade.

## V.2 O programa de aquisição

O programa de aquisição que utilizamos (aquisicao) está listado abaixo. O parâmetro de entrada `tmax` determina a duração da aquisição de dados em milissegundos. O programa lê continuamente o tempo e o estado do botão A1, que corresponde ao pino da interface de jogos ligado ao fotosensor. A cada leitura os resultados são colocados nas variáveis `t` e `a1`, que em seguida são escritas em um arquivo chamado `dados.txt`. O valor de `a1` é 0 quando o sensor está iluminado e 16 quando ele está no escuro.

```
aprenda aquisicao :tmax
abraparaescrever "dados.txt
mudeescrita "dados.txt
atribua "t0 tempomili
enquanto [:t < :tmax] ~
[
```

```

atribua "t tempomili-:t0
atribua "a1 (bite 16 portaentradab 513)
(escreva :t :a1)
]
mudeescrita []
fechearq "dados.txt
escreva [*** fim da aquisição ***]
fim
    
```

Para obter o período e amplitude de cada oscilação do pêndulo devemos analisar os dados contidos no arquivo gerado pelo programa de aquisição. Isto é feito pelo programa *analise*, mostrado no Apêndice, que calcula o período e a velocidade angular a cada passagem do pêndulo pela posição de equilíbrio.

### V.3 Resultados experimentais

Mostramos a seguir os resultados de um experimento em que o movimento do pêndulo foi acompanhado durante 4 minutos. A Fig.9 mostra a velocidade angular máxima  $\Omega_0$  como função do tempo. O amortecimento da oscilação causado pelas forças de atrito é bem visível. O período das oscilações como função do tempo está apresentado na Fig.10. A diminuição do período reflete a sua dependência na amplitude que, como vimos, é gradativamente reduzida pelos atritos. A estabilização do período que ocorre após algum tempo corresponde à entrada do pêndulo no “regime de pequenas oscilações”.

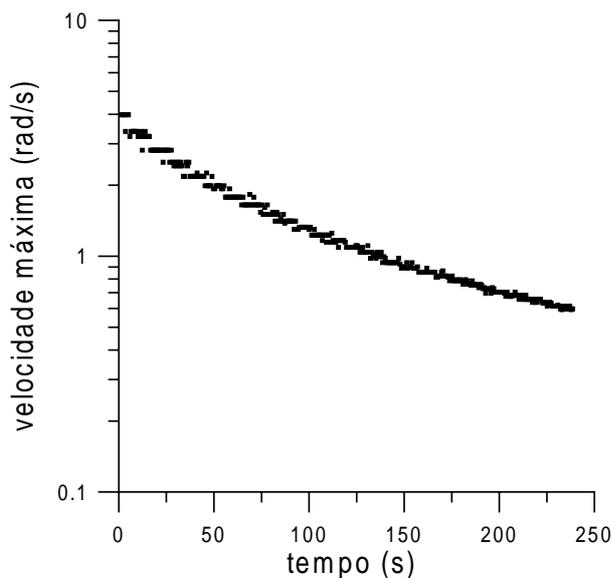


Figura 9. A velocidade angular máxima como função do tempo.

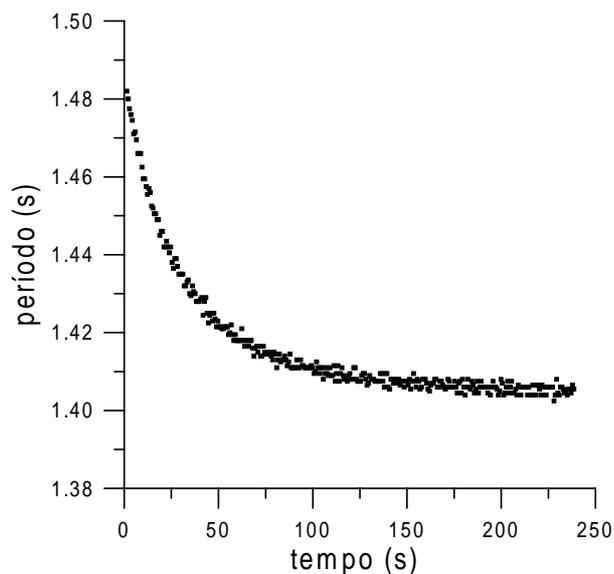


Figura 10. O período como função do tempo.

A Fig.11 mostra como o período depende da velocidade angular máxima. As oscilações de maior amplitude têm  $\Omega_0 \approx 4$  rad/s, o que corresponde a  $\Theta_{\max} \approx 50^\circ$ . Para estas amplitudes o período é cerca de 0,08 s maior que para as pequenas oscilações, um aumento da ordem de 5%.

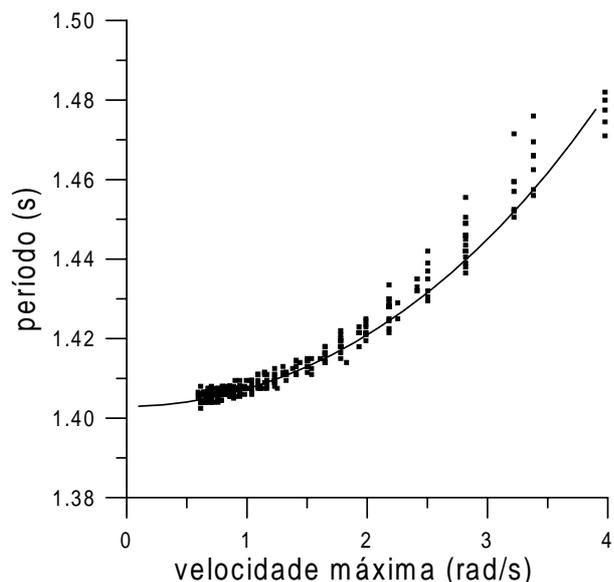


Figura 11. O período como função da velocidade angular máxima. A curva contínua é o resultado do cálculo descrito na seção V.4.

Para as pequenas oscilações sabemos que o período é dado por

$$T = 2\pi \sqrt{\frac{R_G}{g}}$$

O pêndulo que utilizamos tem um raio de giração  $R_G = 48,8 \pm 0,2$  cm. Usando  $g = 978,8$  cm/s<sup>2</sup> para a

aceleração gravitacional no Rio de Janeiro (segundo o Observatório Nacional) obtemos  $T = 1,403 \pm 0,003$  s. A Fig. 11 mostra que este cálculo está em bom acordo com o resultado experimental a baixas velocidades (pequenas amplitudes).

#### V.4 Período do pêndulo a grandes amplitudes

A Fig.11 mostra algo que dificilmente é medido em um laboratório didático convencional: a variação do período de um pêndulo com a amplitude da oscilação. Comparar esta medida com a previsão da mecânica Newtoniana é instrutivo. Um cálculo simples do período de grandes oscilações pode ser realizado resolvendo numericamente a equação de movimento do pêndulo. Isto é feito no programa Logo listado a seguir, `pendulo`, que calcula o período (em segundos) como função da velocidade angular máxima, dada pelo parâmetro de entrada `omega0` (em rad/s). No programa a variável `x` é o comprimento do arco descrito pelo pêndulo a partir do ponto de equilíbrio, e `v` e `a` são a velocidade e aceleração correspondentes. O efeito das forças de atrito sobre o período foi ignorado por ser muito pequeno no nosso caso, mas pode ser incluído facilmente no programa se for necessário.

```
aprenda pendulo :omega0
atribua "dt 0.0001 ;salto de tempo (s)
atribua "g 978.8 ;acel. grav. (cm/s2)
atribua "rg 48.8 ;raio de giração (cm)
atribua "t 0 ;instante inicial
atribua "x 0 ;posição inicial
atribua "v :omega0*:rg ;veloc. inicial
façaenquanto [passo] [:x > 0]
atribua "período 2*(t-x/v)
escreva (lista :omega0 :período)
fim
```

```
aprenda passo
atribua "a -:g*(senrad :x/:rg)
atribua "v :v + :a*:dt ;nova velocidade
atribua "x :x + :v*:dt ;nova posição
atribua "t :t + :dt ;novo tempo
fim
```

A curva que está na Fig.11 foi obtida com o programa acima. O cálculo reproduz bastante bem o comportamento dos dados experimentais. Observe que o cálculo do período no programa `pendulo` utiliza apenas conceitos que são familiares no ensino médio (essencialmente  $\Delta x = v \Delta t$  e  $\Delta v = a \Delta t$ , veja a rotina `passo`). Este é um bom exemplo de como os computadores podem facilitar a modelagem matemática de fenômenos físicos, tornando-a acessível a alunos que não conhecem cálculo.

## VI Comentários finais

Neste trabalho nós discutimos a utilização da porta de jogos de um PC/Windows como interface para aquisição de dados. O sistema apresentado tem as seguintes características:

- Pode ser montado sem fazer modificações no computador.
- Utiliza como sensores componentes eletrônicos simples e fáceis de encontrar.
- É fácil de compreender e implementar.
- É versátil, podendo ser usado em muitos experimentos diferentes.
- É bastante rápido, pelo menos para os objetivos de um laboratório didático, alcançando resoluções temporais da ordem de milissegundos.
- Tem custo próximo de zero se o computador já estiver disponível.
- Os programas de aquisição e análise de dados podem ser todos escritos em Logo.

O sistema pode ser usado em laboratórios didáticos do ensino médio, em condições tais que *tanto alunos quanto professores são capazes de compreender e controlar todas as etapas do processo de aquisição e análise dos dados*. Isto se deve não apenas à simplicidade da interface, mas principalmente ao fato de que os programas de aquisição e análise são inteiramente escritos em uma linguagem acessível como Logo.

A aquisição de dados pela porta de jogos não representa, obviamente, a única forma de utilização do computador em um laboratório didático. Além dos kits comerciais já mencionados (alguns até usam a porta de jogos), inúmeras outras propostas existem, com diferentes graus de sofisticação e custo. Para alguns exemplos, veja as referências [2, 14, 15, 16, 17, 18]. Qualquer que seja a proposta, parece claro que ao introduzir computadores no laboratório didático passamos a ser capazes de observar uma variedade maior de fenômenos, e a analisá-los com mais facilidade. Se somarmos a isto o potencial que o computador tem enquanto instrumento de modelagem matemática, vemos que estas máquinas podem ter um impacto muito positivo no ensino de física, possibilitando uma ligação entre experimento e teoria que raramente é alcançada com outros meios.

Agradecemos o apoio dado pelo Laboratório Didático do Instituto de Física da UFRJ (LADIF), e pelos Profs. Artur Chaves e Susana de Souza Barros.

## Apêndice

Neste apêndice mostramos o programa *analise*, que utilizamos para analisar os dados obtidos na experiência com o pêndulo (seção V). Os dados são lidos no arquivo *dados.txt*, gerado pelo programa aquisição (ver seção V.2). O programa de análise calcula o instante  $t$  em que ocorre uma passagem do pêndulo pela posição de equilíbrio, e o período  $T$  e velocidade angular  $\Omega_0$  nesta passagem. Para isto o programa localiza os instantes em que o sinal na porta deu os “saltos” mostrados na Fig.8, coloca esses tempos em uma lista (*saltos*), e a partir desta lista calcula os valores de  $t$ ,  $T$  e  $\Omega_0$  (variáveis *t*, *período* e *omega*). Uma tabela com os resultados é escrita no arquivo *tpo.dat*. A variável *fator* usada para calcular  $\Omega_0$  corresponde a  $D/R$  (ver seção V.1). Todos os tempos são convertidos para segundos.

```

aprenda analise
atribua "fator 3.3/48.8 ; valor de D/R
escreva [*** aguarde ***]
;
; >>> Cria lista com tempos de salto
atribua "saltos []
abraparaler "dados.txt
mudeleitura "dados.txt
atribua "dado leialista
atribua "t0 primeiro :dado
atribua "x0 último :dado
atribua "xi :x0
atéque [éfimarq] ~
[
  atribua "dado leialista
  atribua "t1 primeiro :dado
  atribua "x1 último :dado
  teste :x1=:x0
  sefalso ~
  [
    atribua "saltos juntenofim :t1 :saltos
    atribua "t0 :t1
    atribua "x0 :x1
  ]
]
atribua "xf :x1
mudeleitura []
fechearq "dados.txt
;
; >>> Primeiro salto deve ser 0->16
se (:xi=16) [atribua "saltos sp :saltos]
; >>> Último salto deve ser 16->0
se (:xf=16) [atribua "saltos su :saltos]
;
; >>> Calcula tempo, período e omega
abraparaescrever "tpo.dat
mudeescrita "tpo.dat
atribua "nsaltos numelem :saltos

```

```

para [i 1 :nsaltos-6 2] ~
[
  atribua "t1 elemento :i :saltos
  atribua "t2 elemento :i+1 :saltos
  atribua "t3 elemento :i+2 :saltos
  atribua "t4 elemento :i+3 :saltos
  atribua "t5 elemento :i+4 :saltos
  atribua "t6 elemento :i+5 :saltos
  atribua "período (:t5+:t6-:t1-:t2)/2000
  atribua "omega :fator/(:t4-:t3)*1000
  atribua "t (:t3+:t4)/2000
  (escreva :t :período :omega)
]
mudeescrita []
fechearq "tpo.dat
escreva [*** fim da análise ***]
fim

```

## References

- [1] J.C. Alves, *Uma Proposta Pedagógica para Uso do Computador em Ambientes de Ensino Experimental de Física*, Tese de Doutorado, COPPE-UFRJ, 2000
- [2] M.A. Cavalcante e C.R.C. Tavoraro, *Projete Você Mesmo Experimentos Assistidos por Computador: Construindo Sensores e Analisando Dados*, Revista Brasileira de Ensino de Física **22** (2000) 421
- [3] M.L. de Jong e J.W. Layman, *Using the Apple II as a laboratory instrument*, The Physics Teacher **22** (May 1984) 291
- [4] W.M. Gonçalves, A.F. Heinrich e J.C. Sartorelli, *Aquisição de dados com a porta de jogos de computadores Apple*, Revista de Ensino de Física **13** (1991) 63
- [5] J. Fuller, *Science Experimenters' Kit*, <http://www.southwest.com.au/~jfuller/scikit.zip>
- [6] R. Haag, *Utilizando a Placa de Som do Micro PC no Laboratório Didático de Física*, Revista Brasileira de Ensino de Física **23** (2001) 176
- [7] Núcleo de Informática Aplicada à Educação (Nied/Unicamp), *SuperLogo 3.0*, <http://www.nied.unicamp.br>
- [8] G. Mills, *MSWLogo*, <http://www.softronix.com/logo.html>
- [9] J. Fuller, *Games Port Input*, <http://www.southwest.com.au/~jfuller/logotut/games.htm>
- [10] T. Engdal, *Joysticks and other game controllers*, <http://www.hut.fi/Misc/Electronics/docs/joystick/>
- [11] R. Zelenovsky e A. Mendonça, *PC: um guia prático de hardware e interfaceamento*, Editora MZ, 1999
- [12] G.J. Deboo e C.N. Burrous, *Integrated Circuits and Semiconductor Devices*, McGraw-Hill, 1977
- [13] S.M. Sze, *Physics of Semiconductor Devices*, Wiley, 1969
- [14] R.D. Peters, *Experimental computational physics using an inexpensive microcomputer*, Computers in Physics (July/Aug 1988) 68

- [15] P.J. Collings e T.B. Greensdale, *Using the computer as a laboratory instrument*, The Physics Teacher (Feb 1989) 76
- [16] R.V. Ribas, A.F. Souza e N. Santos, *Um sistema de aquisição de dados de baixo custo para o laboratório didático*, Revista Brasileira de Ensino de Física **20** (1998) 293
- [17] D.Fagundes, J. Sartori, T. Catunda e L.A.O. Nunes, *Usando a porta paralela do micro PC*, Revista Brasileira de Ensino de Física **17** (1995) 196
- [18] D.F. Souza, J. Sartori, M.J.V. Bell e L.A.O. Nunes, *Aquisição de dados e aplicações simples usando a porta paralela do micro PC*, Revista Brasileira de Ensino de Física **20** (1998) 413