

Photogate de baixo custo com a porta de jogos do PC

(*Low-cost gameport photogate*)

Guilherme Dionisio e Wictor C. Magno¹

Ciências Exatas e Tecnológicas, Unisinos, São Leopoldo, RS, Brasil

Recebido em 20/7/2006; Revisado em 23/10/2006; Aceito em 17/11/2006

Apresentamos neste trabalho um dispositivo de baixo custo para medidas de intervalos de tempo, velocidades e acelerações, bastante útil no problema de queda livre e em outros experimentos de mecânica. Construímos uma *photogate* usando a porta de jogos do PC para fazer aquisição de dados e para obter a aceleração da gravidade. Obtemos boa concordância dos dados experimentais com um modelo teórico de um corpo em queda livre. Nosso trabalho ressalta uma forma de medir com precisão curtos intervalos de tempo, e abre novas possibilidades de aplicação ao ensino de física.

Palavras-chave: aquisição de dados, porta de jogos.

We present in this work a low-cost device for measurements of time intervals, velocities and accelerations, useful in the free-fall problem and others experiments in mechanics. We have constructed a homemade photogate using the PC gameport to make data acquisition and to obtain the acceleration of gravity. We have obtained good agreement of the experimental data and theoretical model of a free falling body. Our work provide a way to measure short time intervals with good accuracy, and open new possibilities of applications in physics teaching.

Keywords: Data acquisition, gameport.

1. Introdução

Os computadores desempenham hoje um importante papel no mundo moderno em quase todos os campos do conhecimento. Na área de ensino de física, por exemplo, existem inúmeras possibilidades de aplicação dos computadores no controle de processos, na coleta de dados, na aquisição de variáveis físicas, em testes de modelos teóricos, dentre outras utilidades que incrementem de forma eficiente o processo ensino-aprendizagem. Dominar e disseminar as novas tecnologias educacionais que utilizam a informática é um desafio a ser enfrentado pelos novos educadores. Seguindo essa linha de pesquisa, vários artigos foram publicados recentemente, evidenciando a utilização do computador para realização de diferentes experimentos didáticos com materiais de baixo custo, Refs. [1-6].

Como já foi muito divulgado, as diferentes portas de acesso de dados do PC, como a *RS232* (porta serial) e a *CENTRONIX* (porta paralela), são excelentes meios para fazer aquisição de dados e também para controlar equipamentos e outros dispositivos usados em experimentos científicos, [7]. Mais recentemente a porta USB do computador (*Universal Serial Bus*) vem substituindo as portas serial e paralela, em virtude da sua alta taxa de transferência de dados, além da progres-

siva extinção das antigas portas nos microcomputadores modernos. Outro meio de coletar dados em experimentos didáticos pode ser através da placa de som do PC, a qual pode ser útil em medidas que envolvem sinais elétricos de baixa amplitude e em frequências não muito altas, conforme pode ser verificado nas Refs. [3, 8, 9, 10].

Uma forma bastante eficiente para medir-se com o computador intervalos de tempo ou para monitorar o estado de movimento em diferentes experimentos mecânicos envolve a utilização de um dispositivo eletrônico conhecido como *photogate*, uma “porta óptica” ou mais precisamente uma chave acionada por luz ou ainda um sistema de disparo ou de engatilhamento (*trigger*) ativado por luz. Diferentes interfaces comerciais de aquisição de dados (Refs. [11, 12]) oferecem soluções prontas utilizando a *photogate* na determinação de períodos de oscilação para pêndulos, sistemas massa-mola, dentre outras aplicações na área de mecânica.

Na Ref. [13] os autores apresentam duas interessantes aplicações da *photogate* envolvendo medidas de intervalos de tempo em experimentos mecânicos, para um pêndulo amortecido e para um pião girante, usando a porta de jogos da placa de som do computador e o programa *Excel* para leitura dos dados.

¹E-mail: wictor@unisinos.br.

O presente trabalho tem como objetivo complementar os resultados alcançados pelos autores e publicados na Ref. [13], agora aplicando o sistema com a porta óptica de baixo custo na medida da aceleração da gravidade para um objeto em queda livre. Na seção 2 discutimos os detalhes da construção da *photogate* e do sistema de medição da velocidade e da aceleração de queda de um corpo abandonado próximo do solo. Comparamos os resultados obtidos com nosso sistema experimental de baixo custo e com um sistema comercial de aquisição de dados da Vernier [11]. Finalmente na seção 3 apresentamos as conclusões deste trabalho e as perspectivas futuras de aplicação do aparato experimental construído.

2. Construção da *photogate* com a *gameport*

2.1. Porta de jogos

A interface de jogos da placa de som do PC é uma porta de entrada e saída de dados com resolução de 8 bits, cuja leitura e escrita são feitas por um *software* de controle nos endereços hexadecimais *I/O 201h* e *209h*, ou decimal *I/O 513*. Também conhecida como porta do *joystick*, a porta de jogos normalmente é encontrada como um conector de 15 pinos (DB15 fêmea). Cada *byte* enviado para a porta contém a informação de cada uma das oito entradas do dispositivo, sendo quatro entradas digitais e quatro analógicas. Desta forma, o estado de cada uma das entradas pode ser verificado lendo-se o bit correspondente na pinagem de entrada que se deseja (Fig. 1). Os pinos 2, 7, 10 e 14 são as entradas digitais, enquanto que as entradas analógicas estão conectadas nos pinos 3, 6, 11 e 13. Os pinos 1, 8 e 9 do conector DB15 fornecem uma tensão contínua $V_{cc} = +5$ V. Os pinos 4 e 5 são terminais comuns de terra (*ground*). Os pinos 12 e 15 não estão conectados. Para a montagem da *photogate*, utilizamos neste trabalho somente os pinos 1, 2, 3, e 4 da porta do *joystick* (ver Fig. 1), conforme veremos em detalhe adiante.

Comparando-se com as portas serial e paralela, a interface de jogos é uma porta de comunicação “síncrona”, o que significa que o envio de cada *byte* é feito em intervalos regulares, não importando a atividade da porta. Nas entradas serial e paralela a comunicação depende do nível de atividade da porta, como acontece por exemplo com um *mouse* ligado a uma porta serial, que envia dados com maior frequência se este dispositivo estiver em movimento ou se houver cliques em seus botões. Entretanto, se o *mouse* não estiver em uso a comunicação será feita com menor frequência. Estas são as chamadas portas “assíncronas”, nas quais o envio de dados é feito em intervalos irregulares de tempo, dependendo do nível de atividade do dispositivo ligado a elas. A característica “síncrona” da porta de jogos a torna de extrema importância na criação de

rotinas computacionais de aquisição de dados relacionadas com intervalos de tempo, o que será comentado em detalhe adiante.

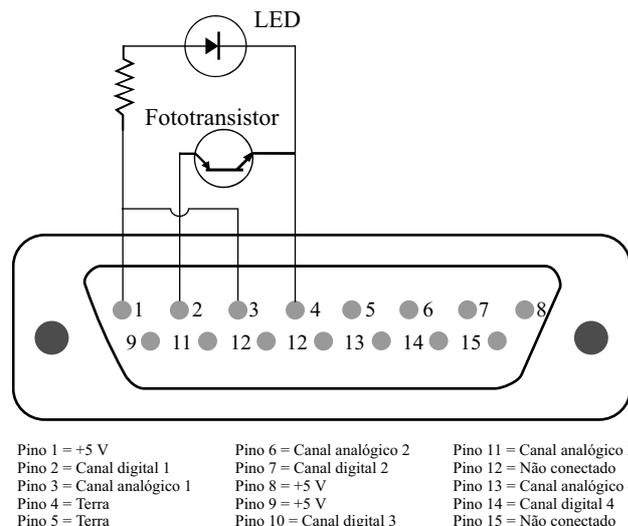


Figura 1 - Montagem de uma porta óptica no conector DB15 da porta de jogos do PC.

2.2. Princípio de funcionamento da *photogate*

O dispositivo *photogate* geralmente é formado por um emissor e um receptor de luz posicionados próximos. O bloqueio do feixe de luz do emissor para o receptor serve para marcar intervalos de tempo que podem ser medidos em eventos sucessivos, disparados ou engatilhados (*triggering*) pela porta óptica. Normalmente são usados LEDs (*light-emitting diode*) ou lasers de diodo como elementos emissores, enquanto que um fototransistor é usado como detector, funcionando como um resistor variável cuja impedância muda em função da intensidade da luz detectada pelo receptor. Outro tipo de *photogate* muito comum utilizada para posicionamento nos *mouses* dos computadores emprega circuitos com amplificadores operacionais de gatilho do tipo Schmitt (*Schmitt-triggered photogate*). Comumente são utilizados osciladores *timer* 555 e elementos disparadores do tipo SCR (*Silicon Controlled Rectifier* = Retificador Controlado de Silício) para contagem de eventos de disparo. Toda vez que um objeto passa através da porta óptica e interrompe a passagem da luz do emissor para o detector, a diferença de potencial através do fototransistor aumenta e produz um pulso quadrado pelo oscilador 555, servindo para disparar o dispositivo SCR. Ao receber o pulso no terminal de disparo, o SCR comporta-se como um diodo, conduzindo uma corrente elétrica quando estiver polarizado diretamente.

O funcionamento das entradas digitais da porta de jogos é bastante simples: uma chave abre ou fecha um circuito, alterando o estado lógico de um bit entre os estados baixo (0) e alto (1). Uma variação deste método de entrada de dados pode ser feita com o uso da porta óptica, cujo princípio de funcionamento baseia-se na

emissão e recepção de um feixe luminoso para abrir ou fechar o circuito. Na Fig. 1 apresentamos a porta óptica construída e adaptada para conexão na porta de jogos do PC. É uma montagem bastante simples e que utiliza componentes eletrônicos de baixo custo encontrados facilmente no mercado. Utilizamos um LED na faixa do infravermelho e um fototransistor TIL-76 como elemento receptor. Na montagem da Fig. 1 tem-se dois circuitos independentes. No primeiro, tem-se o circuito emissor onde o fotodiodo é polarizado diretamente, sendo alimentado através do pino 1 com tensão de +5 V e aterrado no pino 4. Um resistor de 330Ω é ligado em série com o LED para limitar sua corrente a 15 mA para não danificá-lo. No segundo circuito temos o fototransistor receptor, que está ligado ao pino 2 (ou qualquer outra entrada digital) e ligado ao terra, também no pino 4 (poderia ser também o pino 5, como outra alternativa).

Na Fig. 2 vemos a montagem final da porta óptica, empregando-se peças de *LEGO* na montagem mecânica do dispositivo. Normalmente a corrente na entrada digital é de aproximadamente 5 mA e isto mantém o bit no estado 1. Na verdade, a porta retorna o valor 1 ao bit para correntes aproximadamente maiores do que 4 mA. Para correntes elétricas menores que este limite o bit cai para o valor 0. Quando um objeto a ser medido passa entre o emissor e o receptor da *photogate* o fototransistor fica polarizado inversamente enquanto não for iluminado pelo emissor, comportando-se então como uma chave aberta no circuito, pois sua resistência elétrica é elevada, de forma que a corrente na linha digital permanece em 5 mA e o bit no estado 1. No instante em que o feixe de luz do LED é desbloqueado e o receptor é iluminado pelo emissor, o circuito é fechado e uma corrente elétrica da ordem de 1 mA deve ser drenada do pino 2, de modo que a corrente na linha digital torne-se inferior a 4 mA e o bit cai para o estado 0. É importante ainda perceber a ligação entre os pinos 1 e 3 mostrada no circuito da Fig. 1. O pino 3 é uma das entradas analógicas da porta de jogos. É essencial que pelo menos uma destas entradas esteja ligada à fonte de tensão, seja diretamente como foi feito ou por meio de um resistor de qualquer valor, caso contrário a leitura da porta torna-se instável e o estado da linha digital que se deseja ler passará a oscilar aleatoriamente entre os dois estados lógicos, comprometendo assim qualquer aplicação do sensor.

Em resumo, quando o receptor é iluminado pelo feixe de luz proveniente do emissor, uma corrente elétrica é drenada do pino fazendo com que a intensidade da corrente elétrica da entrada digital se reduza a pelo menos 4 mA, o que leva o estado lógico do bit ao valor 0. Quando o feixe luminoso é bloqueado por algum objeto, a corrente elétrica volta ao seu valor máximo alterando o estado lógico do bit para o valor 1. Assim, esta característica de detecção binária pode ser usada como um medidor de intervalos de tempo, o que pode

ser útil na física em experimentos de mecânica, como pêndulos, queda livre, dentre outros.

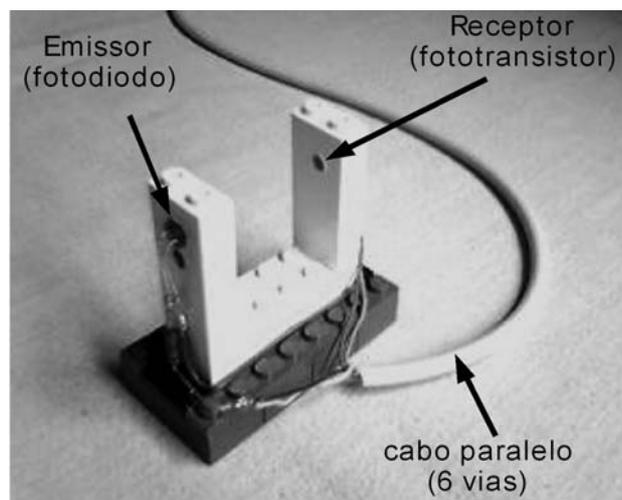


Figura 2 - Montagem de uma porta óptica de baixo custo para experimentos de mecânica.

2.3. Experimento de queda livre

Pode-se simular a queda livre de um objeto pontual passando através de uma porta óptica usando-se uma régua marcada em intervalos regulares conhecida como *picket fence*, [12] (Fig. 3). Isso permite obter-se valores discretos para a contagem do tempo de queda, o que equivale a medir as posições instantâneas de queda do objeto. Com este recurso faz-se uso de um único sensor, ao invés da necessidade de diversos sensores tanto quanto fossem as medidas desejadas. Assim, usando-se uma régua acrílica com fitas escuras coladas igualmente espaçadas entre si e a porta óptica construída, fizemos medidas da velocidade de queda desta régua para posterior análise numérica e obtenção da aceleração da gravidade. A régua utilizada possui 30 cm de comprimento, sendo preenchida com fitas escuras coladas a cada 3 cm. Quando a régua é colocada em queda, o sensor detecta a transição entre as faixas escuras e transparentes da régua, marcando o tempo em que isso ocorre. Com os dados da distância entre as faixas e do intervalo de tempo entre a detecção de duas faixas escuras consecutivas, calcula-se a velocidade média de queda durante o percurso. A velocidade calculada é então relacionada com o instante em que a primeira faixa passa pelo sensor. Pode-se usar também o tempo médio ou o instante em que a segunda faixa atravessa o sensor, pois isto apenas resultará na translação da curva obtida no eixo dos tempos, mas não alterará a declividade da curva $v \times t$, da velocidade em função do tempo, ou seja, a aceleração local da gravidade g , que é o que se deseja medir.

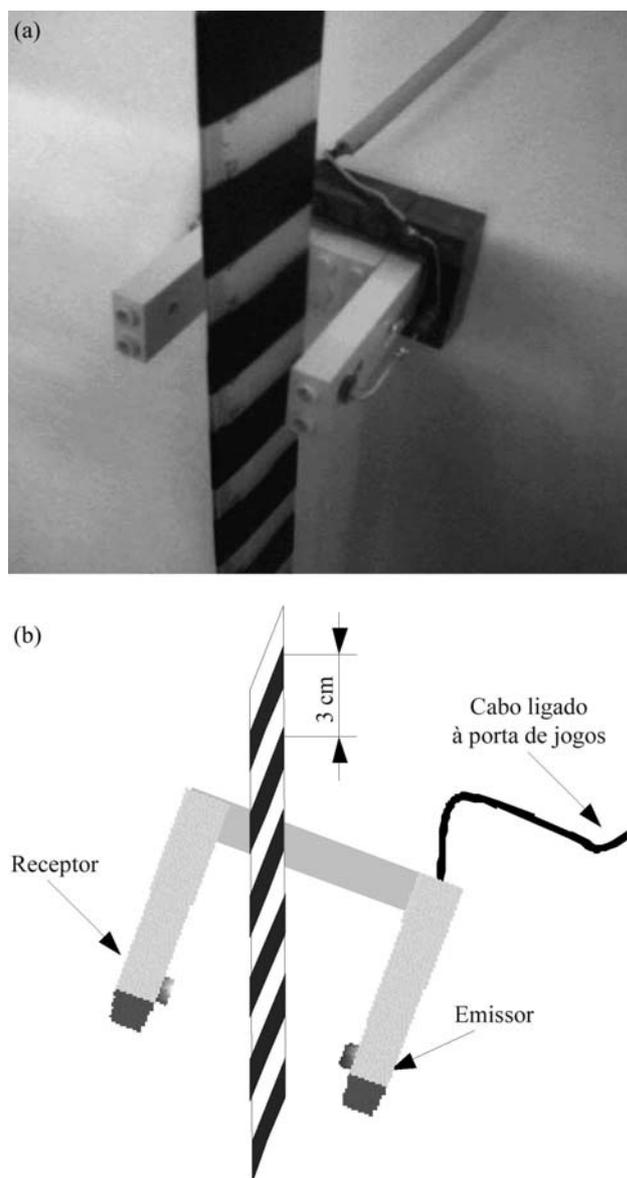


Figura 3 - Experimento de queda livre com a porta óptica construída. (a) Experimento real. (b) Esboço da montagem experimental.

O algoritmo desenvolvido em linguagem C para a coleta dos dados possui uma lógica bastante simples: a cada faixa escura detectada, marca-se o tempo em que o evento ocorre e, a partir daí, tudo se resume a um problema de cálculo de velocidade média. Para a marcação de tempo, foi usado o comando `clock()` que devolve o tempo aproximado de execução do programa. A frequência do relógio interno do computador é de aproximadamente 18.2 Hz, de forma que para obter-se o tempo em segundos é necessário fazer uma conversão deste valor. A escolha por este comando em vez do comando `time()` é por este último retornar o tempo em unidades inteiras de segundos. Como geralmente os tempos envolvidos na medição são de milésimos de segundos (*ms*), o uso deste último co-

mando torna-se inconveniente, devido aos erros de arredondamento. As funções de escrita e leitura das portas I/O são `inp32()` e `oup32()` da biblioteca de link dinâmico `inpout32.dll`.

Ignorando-se a declaração de variáveis, o *kernel* do algoritmo para leitura do sensor óptico construído é o seguinte: ²

```

1      k=0; count=0;
2      tempo1 = clock();
3      xo = ( (inp32) (0x201) & 16 ) / 16;
4      do {
5          x = ( (inp32) (0x201) & 16 ) / 16;
6          k = k++;
7          if (x!=xo & x==1) {
8              count = count++;
9              tempo[count] = k;
10         }
11         xo=x;
12     }
13     while (count<nMedidas);
14     tempo2 = clock();

```

Nas linhas 2 e 14 do código são dadas a partida e a interrupção da contagem do tempo. A diferença entre as variáveis `tempo2` e `tempo1` corresponde ao tempo total de execução do algoritmo. Este valor é convertido em segundos dividindo-o por $(18,2 \times 60 \text{ s})$ (ver código completo no apêndice). Na linha 3, faz-se uma leitura inicial do bit 4 ($2^4 = 16$) da porta de jogos no endereço hexadecimal `h201` (`0x201`). Esta linha do código atribui à variável `x0` o valor 1 se o feixe luminoso está bloqueado e 0 se não está. Na linha 5, uma nova leitura é feita e comparada na linha 7 com a anterior. Se a condição for verdadeira, o programa executa as linhas 8 e 9. Na linha 9, marca-se o tempo em que isso ocorre em unidades de contagem do *clock* que posteriormente será convertida em segundos. Em cada saída do condicional `if` é feita a atualização do valor da variável de comparação `xo` na linha 11. Este ciclo se repete até que, na linha 13, a variável `count` seja igual ao número de medidas `nMedidas` definido pelo usuário (critério de parada do algoritmo). Detalhando a linha 7 do código, vê-se que duas condições são necessárias para que o *looping* seja executado. A condição `x==1` (equivalente a `x = 1`) certifica-se que o feixe está bloqueado. Mas isto não basta para o propósito do algoritmo, pois esta condição é satisfeita durante todo o tempo em que o feixe permanecer bloqueado. O que se quer é que seja detectado o instante em que o evento ocorre. Assim, a condição `x!=xo` (equivalente a `x ≠ x0`) verifica se a leitura anterior de comparação é diferente da leitura atual de teste. Logo, apenas quando as duas condições são satisfeitas, tem-se o instante de bloqueio do sensor.

Como os intervalos de tempo envolvidos no processo são extremamente curtos, sendo da ordem de 10^{-3} do

²O código completo do programa para aquisição de dados é fornecido no apêndice, ao final deste trabalho.

segundo, optou-se por fazer a leitura do *clock* indiretamente por meio de contagem. Na linha 9 do algoritmo é atribuída ao instante em que o feixe luminoso foi bloqueado pela faixa escura da régua a variável `tempo[count]`, que retorna o número k de *looping* do (linha 4) executados, ou seja, o número de leituras da porta até então. Desta forma, ao fim da aquisição a variável k possuirá o número total de leituras feitas durante a execução do programa. Assim, pode-se calcular o intervalo de tempo médio levado para fazer cada leitura e este tempo pode ser considerado uma constante, devido ao fato da porta de jogos ser uma porta de comunicação síncrona como mencionado anteriormente. Com este mecanismo de contagem contorna-se o problema de determinação de intervalos de tempo curtos.

Lembrando o propósito do algoritmo, ele deve fornecer a velocidade, a posição e a aceleração da régua em queda livre em função do tempo. A cada faixa escura detectada pelo sensor, o algoritmo de aquisição marca o instante t contado a partir do início da aquisição em que o evento ocorre, de forma que nas Eqs. (1) e (2) t_n representa o instante em que a n -ésima faixa é detectada pelo sensor e $(t_{n+1} - t_n)$, o intervalo de tempo entre a detecção de duas faixas consecutivas. Estes tempos são salvos pelo algoritmo nas constantes `tempo[count]` e convertidos em segundos como já mencionado. Desta forma, conhecendo-se o espaçamento entre as faixas escuras da régua, padronizado em $\Delta S = 3 \text{ cm}$ (Fig. 4a), e o intervalo de tempo ΔT entre os instantes de detecção destas faixas (Fig. 4b), calcula-se a velocidade média de queda da régua no percurso entre duas faixas consecutivas pela equação

$$\langle v_{n+1} \rangle = \frac{S}{t_{n+1} - t_n}, \quad (1)$$

onde o deslocamento S da régua é dado por múltiplos inteiros do espaçamento entre as faixas, ou seja, 3 cm, 6 cm, 9 cm, e assim por diante. Já para o cálculo da aceleração média, empregamos a equação

$$\langle a_{n+1} \rangle = \frac{\langle v_{n+1} \rangle - \langle v_n \rangle}{t_{n+1} - t_n}. \quad (2)$$

2.4. Análise dos resultados

O modelo teórico para o experimento de queda livre fornece as seguintes equações horárias de movimento para o deslocamento vertical $\Delta y(t)$, para a velocidade de queda $v(t)$ e para a aceleração $a(t)$ de um corpo liberado a partir do repouso ($v_0 = 0$), caindo livremente no campo gravitacional da Terra

$$\Delta y(t) = y(t) - y_0 = \frac{g}{2} \cdot t^2, \quad (3)$$

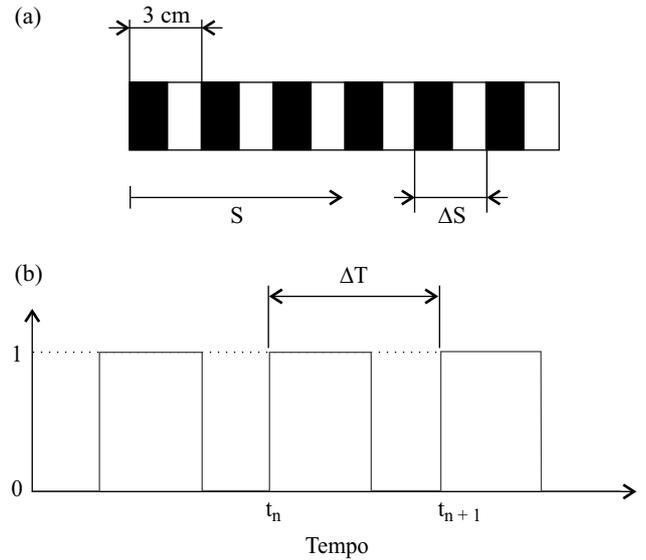


Figura 4 - (a) Régua de referência para medida da aceleração da gravidade em um experimento de queda livre. (b) Estados lógicos detectados pela *photogate* construída, para determinação dos intervalos de tempo de queda livre da régua de referência.

$$v(t) = \frac{d}{dt}y(t) = g \cdot t, \quad (4)$$

$$a(t) = \frac{dv(t)}{dt} = g = \text{constante}. \quad (5)$$

Munidos dos dados fornecidos pelo algoritmo de aquisição é possível traçar os gráficos da velocidade, posição e aceleração da régua em função do tempo. Apresentamos na Fig. 5 os resultados obtidos em um lançamento da régua. A análise numérica dos dados foi feita com um *software* para ajuste de curvas experimentais. Como podemos observar na Fig. 5a a declividade da curva $v(t)$ é constante, representando o valor da aceleração de queda livre da régua em cada instante de tempo, o que neste caso corresponde à aceleração local da gravidade. O valor medido para a aceleração da gravidade foi $g = 9,87 \pm 0,07 \text{ m/s}^2$.

Verifica-se ainda na Fig. 5b que o deslocamento vertical $\Delta y(t)$ de queda da régua ajusta-se perfeitamente bem a um polinômio de grau dois, como previsto pelo modelo teórico, sendo que o coeficiente do termo quadrático obtido do ajuste pelo método dos mínimos quadrados é equivalente a $g/2$, ou seja, a metade do valor da aceleração imprimida à régua pela gravidade. Sendo assim, o ajuste destes dados a uma parábola forneceu para a aceleração da gravidade o valor $g = 9,80 \pm 0,10 \text{ m/s}^2$. O modelo teórico da queda livre de um corpo prevê ainda que a aceleração deste corpo será uma constante no tempo e assim foi demonstrado experimentalmente na Fig. 5c. Neste caso, chega-se ao valor médio $g = 9,80 \pm 0,40 \text{ m/s}^2$. É de se esperar que o desvio padrão aqui seja maior do que

os obtidos nos ajustes anteriores e assim os pontos do gráfico ficam mais dispersos, visto que a aceleração é a derivada da velocidade, de forma que os erros de medida propagam-se ao longo dos cálculos algébricos. Os resultados obtidos com este lançamento já forneceram valores bastante próximos do valor real da aceleração da gravidade. Entretanto, vários outros lançamentos sucessivos e independentes da régua foram realizados, para posterior cálculo do valor médio da declividade das curvas $v(t)$. Os resultados obtidos para g não mudaram muito dos valores apresentados anteriormente.

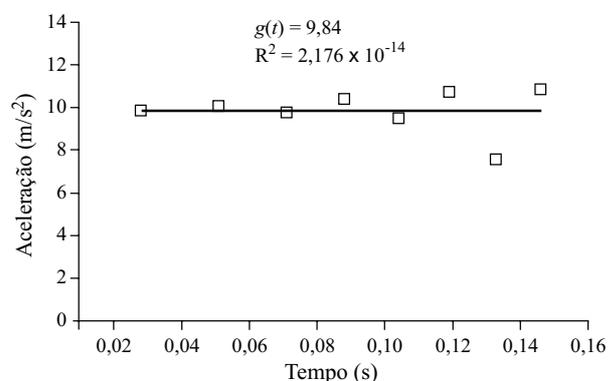
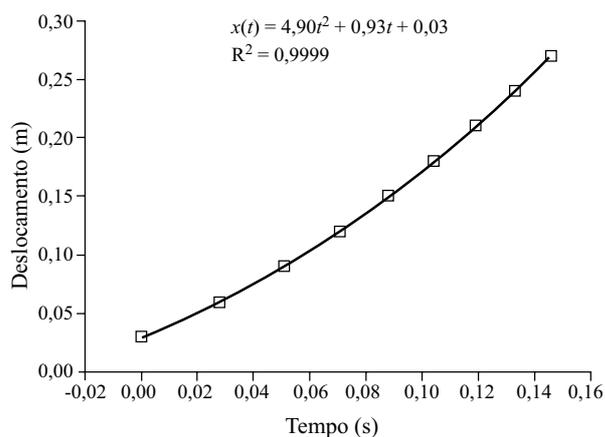
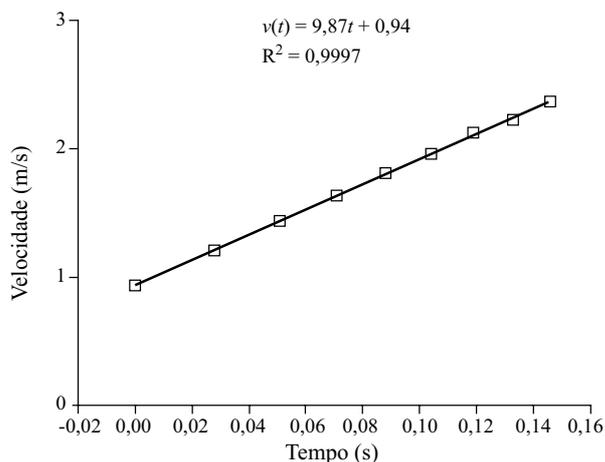


Figura 5 - Resultados do experimento de queda livre. (a) velocidade de queda da régua em função do tempo. (b) deslocamento vertical da régua em função do tempo. (c) aceleração da gravidade em função do tempo.

3. Conclusão

Discutimos neste trabalho o uso da interface de jogos do computador para aquisição de dados em experimentos didáticos. Utilizamos a porta de jogos da placa de som do PC como um conversor analógico/digital de baixo custo, útil como ferramenta de instrumentação de apoio ao ensino de física. Especificamente, construímos uma porta óptica de baixo custo para determinação precisa de intervalos de tempo, a qual pode ser empregada na realização de inúmeros experimentos didáticos em que a medida do tempo desempenha um papel chave na determinação das grandezas físicas medidas com o computador. Para exemplificar o uso da *photogate* de baixo custo construída, montamos um experimento de queda livre em que foi possível medir de forma simples e com boa precisão a aceleração local da gravidade. Diversos experimentos de mecânica envolvendo, por exemplo, medidas de períodos de oscilação em pêndulos podem utilizar o sistema experimental proposto, de forma a facilitar a realização de experimentos didáticos em laboratórios de ensino de física.

Apêndice

Código do programa *photogate.exe*, um *software* escrito em linguagem de programação C para coletar os intervalos de tempo de queda livre de um objeto em um experimento de mecânica. A rotina foi compilada com o software DEV C++ com distribuição livre para plataforma *WINDOWS*. Devem ser carregadas as bibliotecas *stdio.h*, *conio.h*, *stdlib.h*, *time.h* e *windows.h* e declaradas as funções *I/O* como parte da função *main()* na rotina. O uso de outro compilador pode requerer alterações nas bibliotecas carregadas ou em algumas linhas de comando.

```
// photogate.c - V. 1.0 - Unisinos, julho 2006
```

```
int main(void) {
    /*** decl. de var.s e atrib. de vals. ***/

    int porta, x, xo, nMedidas, k, i, count;
    float diametro; double
    delta; clock_t tempo1, tempo2; porta=0x201;
    //endereco da porta de jogos
    nMedidas=10;
    //numero de faixas que a regua possui
    diametro=0.03;
    //espacamento entre as faixas escuras
    double t[nMedidas], instante[nMedidas];
    double velocidade[nMedidas], posicao[nMedidas],
    aceleracao[nMedidas];
    int tempo[nMedidas];
```

```

/** inicio da aquisicao **/

k=0; count=0; tempo1 = clock();
xo = ((inp32)(porta) & 16)/16;
//estado inicial da porta (leitura de comp.)

do {
  x = ((inp32)(porta) & 16)/16;
  //le a porta
  k = k++;
  //conta o numero de looping
  if (x!=xo & x==1) {
    // verifica se o estado alterou
    // e se o feixe foi bloq.
    count = count++;
    //conta quantas vezes mudou o estado
    tempo[count] = k;
    //marca o tempo em func. da contagem de loop
  }
  (oup32)(porta,x);
  // saida na porta para garantir a
  // estabilidade da leitura
  xo=x;
  //faz a leitura atual ser igual a de compar.
}
while (count<nMedidas);
  tempo2 = clock();
/** fim da aquisicao **/

/** transformacao dos tempos **/

delta = (tempo2-tempo1)/(18.2*60*k);
// retorna o tempo de cada leitura
// com precisao de 10 ms
for (i=1; i<=nMedidas; i++) {
  t[i]=tempo[i]*delta;
  //transforma o t[i] em un. de tempo
}

/** criacao do arquivo de dados **/

FILE *arquivo;
arquivo = fopen("data.txt", "w");
//abre arquivo para escrita

velocidade[0]= velocidade[1];
for (i=1; i<nMedidas; i++) {
  instante[i] = t[i] - t[1];
  //instante de deteccao

```

```

  velocidade[i]=diametro / (t[i+1]-t[i]);
  //calculo da velocidade
  posicao[i]=i*diametro;
  //calculo da posicao
  aceleracao[i]=
    (velocidade[i]-velocidade[i-1])/(t[i+1]-t[i])
  //calculo da aceleracao

  fprintf(arquivo, "%.3f\t %.3f\t %.3f\n",
    instante[i], velocidade[i], posicao[i]);
}
fclose(arquivo); //fecha arquivo

```

Referências

- [1] M.A. Cavalcante e C.R.C. Tavolaro, Rev. Bras. Ens. Fís. **22**, 421 (2000).
- [2] M.A. Cavalcante e C.R.C. Tavolaro, Rev. Bras. Ens. Fís. **22**, 247 (2000).
- [3] Erivaldo Montarroyos e Wictor C. Magno, Rev. Bras. Ens. Fís. **23**, 57 (2001).
- [4] C.E. Aguiar e F. Laudares, Rev. Bras. Ens. Fís. **23**, 371 (2001).
- [5] Rafael Haag, Rev. Bras. Ens. Fís. **23**, 176 (2001).
- [6] Alejandro R.G. Ramirez, Milton José Cinelli e Adriano Mansur Irigoite, Rev. Bras. Ens. Fís. **27**, 609 (2005).
- [7] Willis J. Tompkins and John G. Webster, *Interfacing sensors to the IBM PC* (Prentice-Hall, New Jersey, 1987), p. 447.
- [8] Wictor C. Magno e Erivaldo Montarroyos, Rev. Bras. Ens. Fís. **24**, 497 (2002).
- [9] W.C. Magno, A.E.P. de Araujo, M.A. Lucena and E. Montarroyos, Rev. Bras. Ens. Fís. **26**, 117 (2004).
- [10] André L. Bender, Diomar R. Sbardelotto e Wictor C. Magno, Rev. Bras. Ens. Fís. **26**, 401 (2004).
- [11] “Vernier photogate”, <http://www.vernier.com/probes/vpg-btd.html>. Acesso em 20/7/2006.
- [12] “Pasco photogate”, <http://www.pasco.com/products/groups/53-351-1.html>. Acesso em 20/7/2006.
- [13] Jalves S. Figueira e Eliane A. Veit, Rev. Bras. Ens. Fís. **26**, 203 (2004).
- [14] Software desenvolvido em Delphi para leitura da porta de jogos. <http://www.if.ufrgs.br/cref/ntef/index.html>. Acesso em 10/10/2006.