

CBPF-NT-004/89

INTERFACE APPLE PARA INSTRUMENTAÇÃO E CONTROLE DE
EXPERIÊNCIAS - CONTADOR DE PULSOS, "TIMER", CON-
VERSOR DIGITAL-ANALÓGICO, MOTOR DE PASSO E RELES

por

J. Heitor SOUZA, Geraldo R.C. CERNICCHIARO e
J. Thadeu P.D. CAVALCANTE

Centro Brasileiro de Pesquisas Físicas - CBPF/CNPq
Rua Dr. Xavier Sigaud, 150
22290 - Rio de Janeiro, RJ - Brasil

Resumo - Neste artigo descrevemos uma placa de interface para microcomputador tipo Apple II desenvolvida para automatizar sistemas de medidas onde se necessite de contador de pulsos TTL, saída de tensão analógica (com resolução de 12 bits), saídas para controle de motor de passo, acionamento de relés, e "timer" para controle de intervalo de tempo real na realização de tarefas paralelas. Apresentamos também uma aplicação desta placa a um aparelho leitor de Termoluminescência (TL).

Palavras-chave: Interface Apple; Termoluminescência.

1 - INTRODUÇÃO

Na instrumentação e controle de experiências em de laboratórios de pesquisa, existem ainda um grande número de interfaces para estabelecer a interação entre experiências e microcomputadores que, ou não se encontram normalmente no mercado ou são de difícil implementação.

Neste projeto foi desenvolvida uma interface que concentra várias funções comuns a instrumentação e controle de experiências tais como: contador de pulsos TTL, "timer", gerador de tensão analógica com resolução de 12 bits, bem como de controles eletro-mecânicos (motor de passo e relês). Neste projeto, estivemos sempre preocupados com a confiabilidade e a facilidade de implementação deste equipamento, assim como com a disponibilidade de seus componentes no mercado. Escolhemos um microcomputador Apple II com cartão CP/M pela sua disponibilidade e preço, o que lhe garantem um variado mercado de interfaces (ADC por exemplo) e de software.

2 - HARDWARE

A interface é composta de dois módulos interligados por um cabo plano de 24 vias e dois cabos coaxiais. O primeiro módulo, mostrado no esquema da fig. 1, consiste em uma placa compatível com o barramento Apple (50 pinos) e contém uma 8253 (programmable counters), tres 74374 (Latch de 8 bits), um conversor Digital- Analógico 7541 da Bechmam e alguns circuitos integrados TTL para decodificação e "driver". O segundo módulo é uma caixa onde estão os 12 conectores tipo

RCA para as saídas das linhas lógicas (4 para o comando do motor de passo e 8 para comando de relês ou outros dispositivos - note que no esquema da fig.1 existem ainda disponíveis, para futuras implementações, 7 bits de controle) e 2 conectores tipo BNC para a entrada de pulsos TTL e saída de tensão analógica.

O 8253 (Intel) reúne 3 contadores de 16 bits (65536 contagens) em um único CI. Pode ser programado para operar em 4 modos diferentes em uma faixa de frequência de 0 a 2MHz. No nosso sistema operamos com dois contadores ligados em cascata (operando no modo 3 e 0) que atuam como contadores de tempo real, cujo final é indicado através de uma saída para dispositivo de sinalização. Com um "clock" de 1 MHz, estes contadores podem cobrir intervalos de tempo de décimos de segundo a horas. O contador restante (programado no modo 0) atua como um contador de eventos em taxas de até 2MHz. Por esta razão sua conexão com o módulo exterior é feita por cabo coaxial.

3 - SOFTWARE BASICO

O software do sistema foi desenvolvido em linguagem Turbo Pascal, em linguagem de máquina do 6502 e do Z80, podendo ser compreendido em seus diversos módulos. Consiste basicamente em se escrever os comandos de controle como tensão do DAC, modo de operação do contador, sequência de controle do motor de passos, posição dos relês e leitura dos módulos contadores, nos seus respectivos endereços.

A rotina de controle do DAC é apresentada a seguir:

```

procedure dac(v:integer);

const
  LSB_DAC=$E0A4;
  MSB_DAC=$E0A8;

begin
  mem[MSB_DAC]:= v div 256; { 4 bits mais significativos }
  mem[LSB_DAC]:= v mod 256; { 8 bits menos significativos }
end;

```

Na rotina de Controle do Motor de Passo e Relés temos um único endereço de escrita, pois um mesmo "latch" é usado para este controle. Usamos então uma máscara para que as duas funções fossem operadas independentemente. O comando dos relés é do tipo liga/desliga, e o comando do motor de passo, menos simples, requer que se leve em consideração a sequência correta de pulsos de energização das bobinas. O motor por nós utilizado (1,8 graus/passos) necessita de 4 bits para realizar uma sequência completa, deste modo restaramos neste "latch", 4 bits para o acionamento de relés.

No projeto implementado tivemos a seguinte configuração: BYTE DE COMANDO = RRRMMMMR, onde cada bit R comanda um relé e os bits M comandam o motor, segundo seus níveis lógicos. Assim para se comandar o motor sem afetar os relés deve-se mascarar (11100001 Hex=255) o conteúdo do "latch" para então se acrescentar o byte correspondente ao passo da sequência de movimentação. Isto foi feito da seguinte forma:

```

Procedure Motor(Giros:Integer); { sentido e no. de passos }

Const
  EndMotor=$E0AC; { endereço alocado para o motor }
  PassoH:Array[1..4] of Byte=(18,24,12,6);
  { sequência de movimentação - sentido horário }

```

```

PassoA:Array[1..4] of Byte=(18,6,12,24);
  { sequência de movimentação - sentido anti-horário }
Retardo=10;           { intervalo entre passos do motor }
Posicao:Byte=18;      { guarda posição atual do motor }
MascaraMotor=225;    { máscara para motor }

Var
  Passo:Array[0..3] of Byte;
  Inicio,Final,i:Integer;

Begin
  If Giros>0 then { transfere p/ Passo sequência real }
    For Inicio:=0 to 3 do
      Passo[Inicio]:=PassoH[Inicio]
    Else
      For Inicio:=0 to 3
        Passo[Inicio]:=PassoA[Inicio];
      Inicio:=0;           { localiza índice inicial }
      While Posicao<>Passo[Inicio] do
        Inicio:=Inicio+1;
      Final:=Inicio+Abs(Giros);
      For i:=Inicio to Final do begin
        Mem[EndMotor]:=Mem[EndMotor] And MascaraMotor +
          Passo[i Mod 4];
        Delay(Retardo)
      End;
      Posicao:=Passo[Final Mod 4]           { posição final }
End;

```

O acionamento dos relés é executado invertendo-se seu estado lógico, ou seja, desliga se estiver ligado e vice-versa, o que foi feito da seguinte forma:

```

Procedure Rele(n:Integer);           { aciona o n-ésimo relé }

Const
  EndRele=$E0AC;           { endereço alocado p/ os relés }
  Rele:Array[1..4] of Byte=(1,32,64,128);
                                { máscaras p/ relés }

Begin
  Mem[EndRele]:=Mem[EndRele] XOR Rele[n]
End;

```

A rotina de programação dos módulos contadores consiste em inicializa-los, escrevendo nos registros de comando os respectivos modos de operação: modo 0 (Interrupt on Terminal Count) para o contador 0 que executa a contagem de pulsos, modo 3 (Square Wave Rate Generator) para o

contador 1 e modo 0 para o contador 2 que combinados fazem a contagem de intervalo de tempo real, usando a base de tempo do micro. Devido a características próprias das linhas de escrita do barramento do Apple esta operação foi executada em linguagem de máquina do 6502 e encontra-se esquematizada no diagrama de blocos da fig.2. Os resultados de contagens (pulsos ou timer) são obtidos lendo-se as posições de memória \$9300 e \$9301.

Rotina de inicialização dos contadores em linguagem do 6502.

```
Procedure Load6502(Com1,EndCont,LSB,MSB:Byte);
```

```
Const
```

```
  A6502:Array[0..15] of Byte=( $A9,0,$8D,$A3,$C0,$A9,0,$8D,  
                                0,$C0,$A9,0,$8D,0,$C0,$60 );
```

```
Var
```

```
  i:Integer;
```

```
Begin
```

```
  For i:=0 to 15 do  
    Mem[$9200+i]:=A6502[i];  
  Mem[$9201]:=-Com1;  
  Mem[$9206]:=-LSB;  
  Mem[$9208]:=-EndCont;  
  Mem[$920B]:=-MSB;  
  Mem[$920D]:=-EndCont;
```

```
End;
```

Rotina de leitura dos contadores em linguagem do 6502.

```
Procedure LoadCont;
```

```
Const
```

```
  B6502:Array[0..17] of Byte=( $A9,$80,$8D,$A3,$C0,$AD,$A2,  
                                $C0,$8D,$00,$A3,$AD,$A2,$C0,$8D,$01,$A3,$60 );
```

```
Var
```

```
  i:Integer;
```

```
Begin
```

```
  For i:=0 to 17 do  
    Mem[$9210+i]:=B6502[i]
```

```
End;
```

Rotinas que ativam o 6502 via CP/M em linguagem de máquina do Z80.

```
Procedure Inicia8253;
Begin
  InLine($21/$00/$A2/$22/$D0/$F3/$2A/$DE/$F3/$77)
End;
```

Rotina de Leitura e Acionamento dos Contadores

```
Procedure LeContador;

Begin
  InLine($21/$10/$A2/$22/$D0/$F3/$2A/$DE/$F3/$77)
End;
```

4 - APLICAÇÃO A UM APARELHO LEITOR DE TL

A interface apresentada neste trabalho foi testada num aparelho leitor de Termoluminescência que será apresentado em detalhes num outro artigo. Neste aparelho o DAC comanda a velocidade de aquecimento da amostra (linear no tempo) , o contador de pulsos TTL faz a contagem de ftons emitidos pela amostra, o motor de passos executa: tanto a troca de filtros óticos que se colocam entre a amostra e o tubo fotomultiplicador, como o posicionamento de uma lâmpada padrão para calibração automática de todo o sistema de contagem de ftons, e finalmente um relé comanda a partida e o término do aquecimento. A temperatura atual da amostra pode ser calculada pelo tempo decorrido depois da partida, ou medida diretamente através de um ADC. Esta segunda alternativa nos parece mais interessante, pois também monitora o correto funcionamento do sistema de aquecimento, e assim sendo empregamos um conversor comercial de 12 bits, o AM-13 da Anamed. Na fig.3 apresentamos uma curva de TL x T obtida através da interface descrita neste trabalho, em

acordo com os resultados de Durrani et al, 1977.

5 - CONCLUSAO

A interface vem apresentando um desempenho dentro das características desejadas, mostrando a viabilidade e a confiabilidade deste sistema, econômico, de fácil implementação e operação. Suas características gerais sugerem que este projeto possa ser aplicado a uma série de outras situações comuns a física experimental.

6- AGRADECIMENTOS

Os autores desejam expressar seu agradecimento ao prof. Alexandre Leib Grojsgold, do Laboratório Nacional de Computação Científica-CNPq, por importantes esclarecimentos a respeito do microprocessador 6502 e ao prof. Gustavo F. P. Figueiredo por contribuições no desenvolvimento do software e do hardware deste trabalho.

LEGENDAS:

- Fig.1** Esquema da interface: contadora de pulsos, DAC, Timer, controladora de motor de passos e acionadora de relés.
- Fig.2** Fluxograma do Programa de Acionamento dos Contadores
- Fig.3** Curva de TL X Temperatura obtida para o Quartzo irradiado sob raios-X.

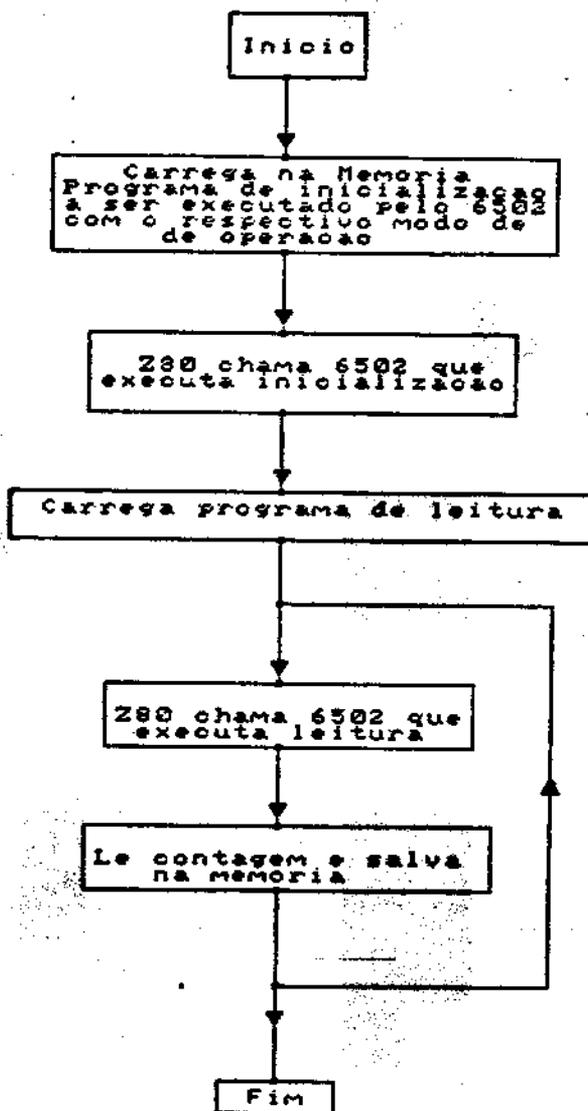


fig. 2

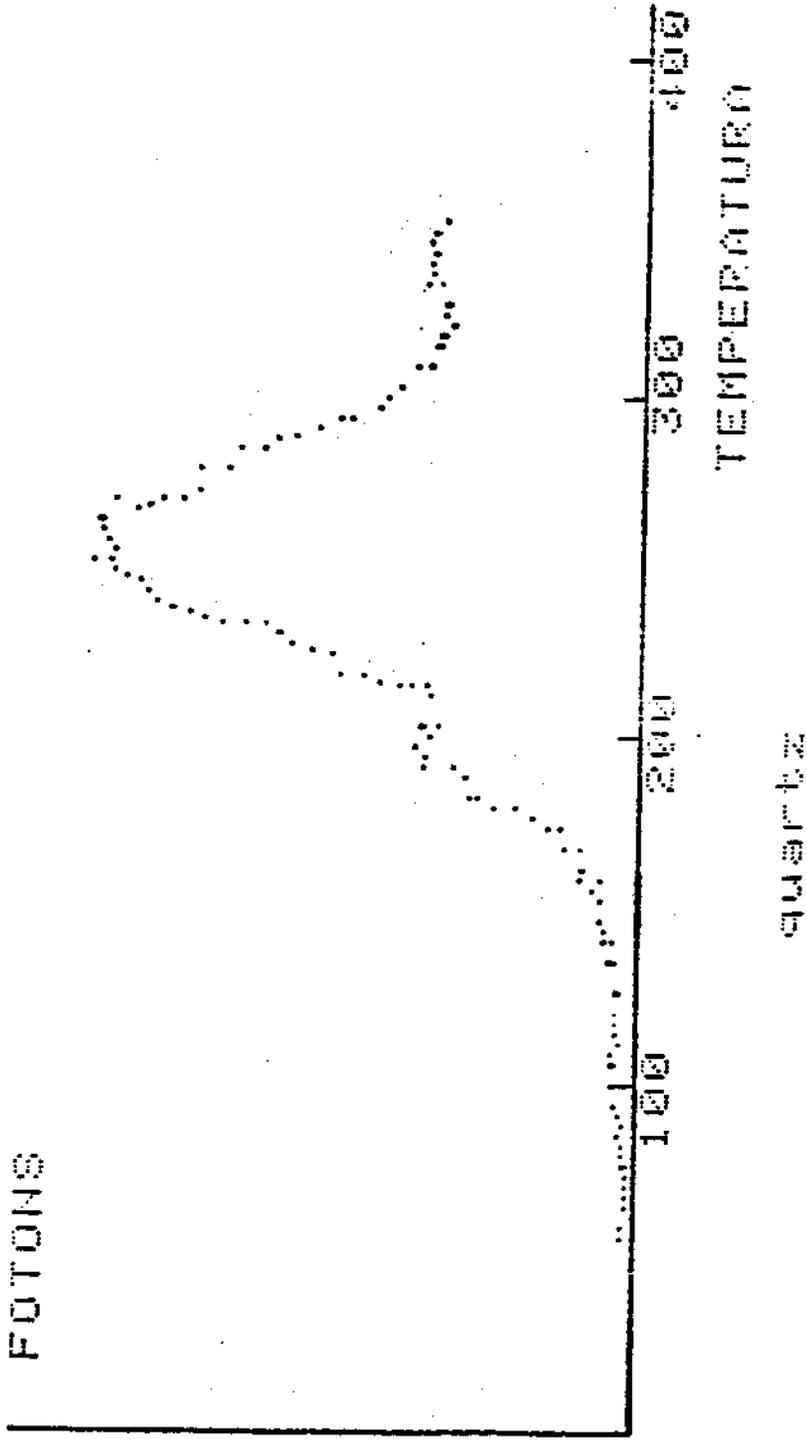


fig. 3

BIBLIOGRAFIA

- 1) INTEL 1980 - COMPONENT DATA CATALOG
- 2) Poblet J. M. 1980, Interconexion de Perifericos e Microprocessadores. Bacharel Editores, Barcelona, Espanha.
- 3) TEXAS - The TTL Data Book, Texas Instruments Corporation, (1980).
- 4) Jonathan A. Titus, David G. Larsen e Christopher A. Titus (1981). Apple Interfacing. Howard W. Sams & Co., Inc., USA.
- 5) Durrani, S.A., Khasal, K.A.R., McKeever, S.W.S. & Riley, R.J. (1977). Radiat. Eff. 33,237.

Abstracts - In this work we describe an interface for a microcomputer Apple, which has been developed to automatize measurement systems where a TTL pulse counter, an output to analogical signal (resolution of 12 bits), output to stepmotor control, relays control and a timer are required. An application to a Thermoluminescence system is also presented.

Palavras-chave: Apple interface; Themoluminescence.