O SEU MICRO E O MUNDO EXTERNO

Neste livro — inédito deste gênero no Brasil — através de projetos claros, funcionais e com componentes facilmente encontrados no mercado nacional, o autor demonstra como é fácil acessar o mundo "externo" do microcomputador.

Partindo de um simples circuito básico, o hobbista pode montar seus mais diversificados periféricos, seja no campo dos jogos, como por exemplo luzes seqüenciais, tiro ao alvo etc., ou seja no campo profissional, como gravador de EPROM, secretária eletrônica, analisador lógico etc.

Um livro que abre novos horizontes para o uso do seu micro!

O autor desenvolvendo um programa no seu TK 85, instalado numa caixa de fibra de vidro da Fa. SPEED/MG.

Bernhard Wolfgang Schön escreveu além desta obra diversos artigos em revistas nacionais e realizou várias palestras sobre a informática e microcomputadores. Com este livro foi eliminada uma lacuna de falta de livros específicos sobre hardware no mercado nacional.

para seu ZX81 • TK82 • TK83 • TK85 • NE-Z8000 • CP200 • TK2000 • APPLE...

EDITORA MODERNA

Editora Nleph

Nleph

O SEU MICRO E O MUNDO EXTERNO

O SEU

MICRO

E O MUNDO EXTERNO

BERNHARD WOLFGANG SCHÖN

periféricos sensacionais!

para seu ZX81 • TK82 • TK83 • TK85 • NE-Z8000 • CP200 • TK2000 • APPLE...

EDITORA MODERNA

Editora Nleph

TABELA DE CONVERSÃO

HEXADECIMAL DECIMAL

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
2	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
3	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
4	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
5	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
6	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
7	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
8	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
9	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
A	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
B	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
C	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
D	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
E	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
F	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

LITEC
LIVRARIA EDITORA TÉCNICA LTDA
Rua dos Timbiras, 257 - CEP: 01208 - São Paulo
Caixa Postal 30 869 - Tel.: 222-0477
REF. PREÇO

O SEU MICRO E O MUNDO EXTERNO

Este livro foi editado, pela
Aleph PUBLICAÇÕES E
ASSESSORIA PEDAGÓGICA LTDA.

Os programas
foram digitados e testados
no NÚCLEO DE ORIENTAÇÃO DE ESTUDOS
Depto. de Cursos de Computação
Av. Brig. Faria Lima, 1451 - conj. 31
01451 São Paulo - SP (813-4555)
sob a coordenação
da professora Betty Fromer Piazzi

Núcleo de orientação de estudos

CIP-Brasil. Catalogação-na-Publicação
Câmara Brasileira do Livro, SP

S392s
Schön, Bernhard Wolfgang, 1954-
O seu micro e o mundo externo
Bernhard Wolfgang Schön. -
São Paulo : A leph, 1985.
1. Microcomputadores I. Título.

85-0028
17. CDD-651.8
18. -001.6404

Índices para catálogo sistemático:
1. Microcomputadores : Processamento de dados
651.8 (17.) 991.6404 (18.)

JCA
20.12.86

O SEU MICRO E O MUNDO EXTERNO

BERNHARD WOLFGANG SCHÖN

EXPEDIENTE

Coordenação editorial:
Pierluigi Piazzi.
Digitação e revisão técnica:
Roberto Bertini Renzetti
Projeto visual e produção gráfica:
Hugo Sergio Faleiros V. V. E.
Ilustrações:
Hindenburgo Alencar
Organização editorial:
Rosana de Angelo
Revisão e copy-desk:
Lúcia Kairovsky
Produção:
Rosa Kogan Fromer
Ilustração da capa:
CADPRESS
T.(011)870155

Aleph

EDITORA MODERNA

ÍNDICE

OBS.: ERRATA da Pag. 22 impressa na última página do livro

INTRODUÇÃO

A rápida popularização dos micro computadores conduziu uma grande parcela de leigos ao fascinante mundo da informática. A maioria deles tomou contato com o micro externamente como usuário, aprofundando-se no estudo do "software".

Uma minoria, porém, quer entender as "entranhas" de seu computador, abordá-lo do ponto de vista do "hardware".

O micro computador é um cérebro que lê e escreve: dotá-lo de olhos, braços, ouvidos, ligá-lo ao mundo externo é fornecer um corpo a uma mente.

Esta possiblidade fascinante nos convenceu a editar esta obra pioneira de Bernhard Schön, para divulgar as enormes (e muitas vezes insuspeitas) possibilidades que seu pequeno micro esconde.

Pierluigi Piazzi

PREFÁCIO

Tenho certeza, meu caro leitor, que você é ou será mais um dos usuários de microcomputadores, que se preocupa com os recursos que o micro oferece. Mesmo não pondo em prática todos os projetos deste livro, você diferencia-se da grande maioria de usuários, que se limita apenas em rodar programas já prontos nos seus computadores.

Na verdade existem, em princípio, três tipos de usuários:

Usuário A — ele compra e troca programas desenvolvidos por terceiros, sem entender o que se passa dentro do computador. Para ele, o seu micro não passa de uma caixa preta, que serve somente para rodar programas prontos.

Usuário B — este já se preocupa com as linguagens de programação existentes, geralmente o BASIC e o ASSEMBLY, e começa a desenvolver seus próprios programas, específicos para suas necessidades. Este usuário sabe que o único limite na programação (e com isto a capacidade do micro de realizar as tarefas mais diversificadas) é a criatividade do próprio programador. Na prática é ele quem domina o computador, enquanto que no caso do usuário A, é o computador quem o domina.

Usuário C — estes são pessoas como você, que querem abrir a barreira entre o microcomputador e o mundo externo, ultrapassando os limites traçados pelas impressoras, monitores e gravadores.

Normalmente, o usuário C já passou a seqüência do tipo A e B, e não se satisfaz com o mundo de programação — seja por necessidade, estudo ou mera curiosidade.

O objetivo deste livro é demonstrar como é simples acessar o mundo externo, quais são os meios, bem como a apresentação de alguns projetos com suas respectivas rotinas de programação.

Este livro, em princípio, é dedicado aos usuários dos micros ZX81, RINGO R-470, TK82, TK83, TK85, NE-Z8000, CP200 e AS1000, mas, com pequenas alterações, os mesmos projetos também podem ser acoplados aos micros TK2000, linha APPLE, TRS 80, COMMODORE e outros.

UMA PALAVRA PARA OS USUÁRIOS DE MICROS NÃO COMPATÍVEIS COM O SINCLAIR

Como já foi mencionado na introdução, apenas pequenas alterações fazem-se necessárias para acoplar os projetos deste livro ao seu computador.

A hardware de todos os projetos é completamente compatível, com exceção do circuito básico, que foi projetado para os micros TK85 e compatíveis. A diferença principal consiste na área de trabalho, pois nos micros TK85, este circuito trabalha na área de memória entre 8192 e 8195 inclusive. Provavelmente, no seu micro, estes endereços estão sendo usados por uma memória, impedindo assim o funcionamento do circuito básico. Neste caso, terá de escolher uma outra área que não esteja ocupada. Como verá nos capítulos seguintes, a seleção do endereço de funcionamento é feita por um multiplexador, cujas entradas devem ser alteradas de acordo com seu computador. O restante do circuito básico permanece inalterado.

No que se refere aos software, acredito que as rotinas escritas em BASIC facilmente podem ser traduzidas para o BASIC do seu micro. As rotinas ASSEMBLY foram escritas para a CPU Z80, e no final deste livro, todas as rotinas empregadas são listadas com seus respectivos mnemônicos, facilitando assim sua compreensão. Dentro desta lógica deverão ser escritas as rotinas ASSEMBLY para seu computador.

Apesar destas diferenças, creio também que você encontrará neste livro informações e incentivos para montar um ou mais dos projetos apresentados, ou até para criar novas montagens.

O CIRCUITO INTEGRADO 8255 PIO

Todos os projetos neste volume baseiam-se num circuito integrado 8255 PIO, que é uma interface de 8 bits, projetado para trabalhar acoplado em sistemas com o microprocessador 8080 da INTEL. Apesar dos micros do tipo TK85 e semelhantes trabalharem com a CPU Z80, esta interface pode ser facilmente comandada por estes microcomputadores.

Em primeiro lugar, vamos dar uma olhada na arquitetura do 8255:

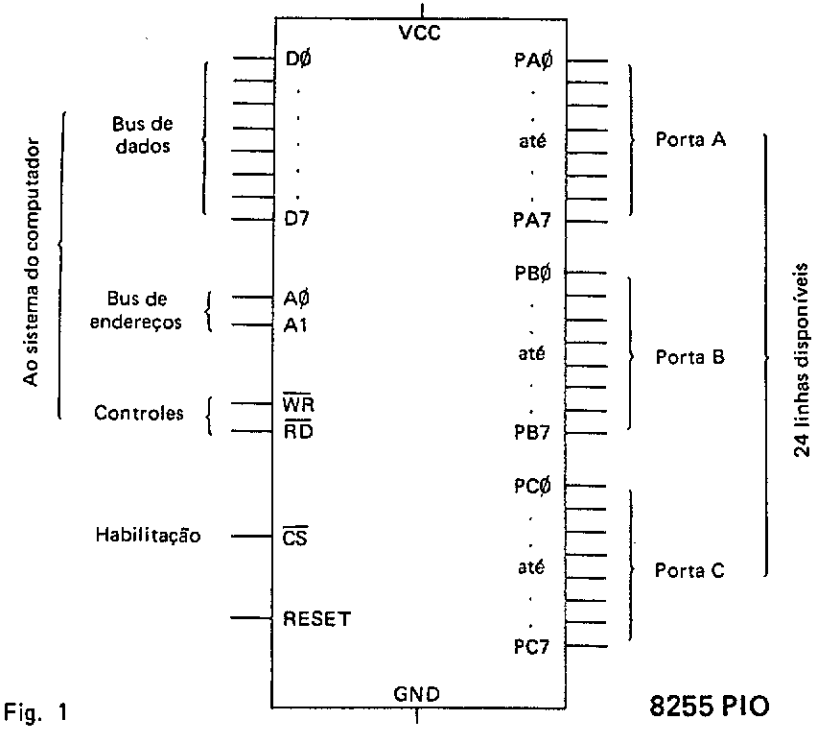
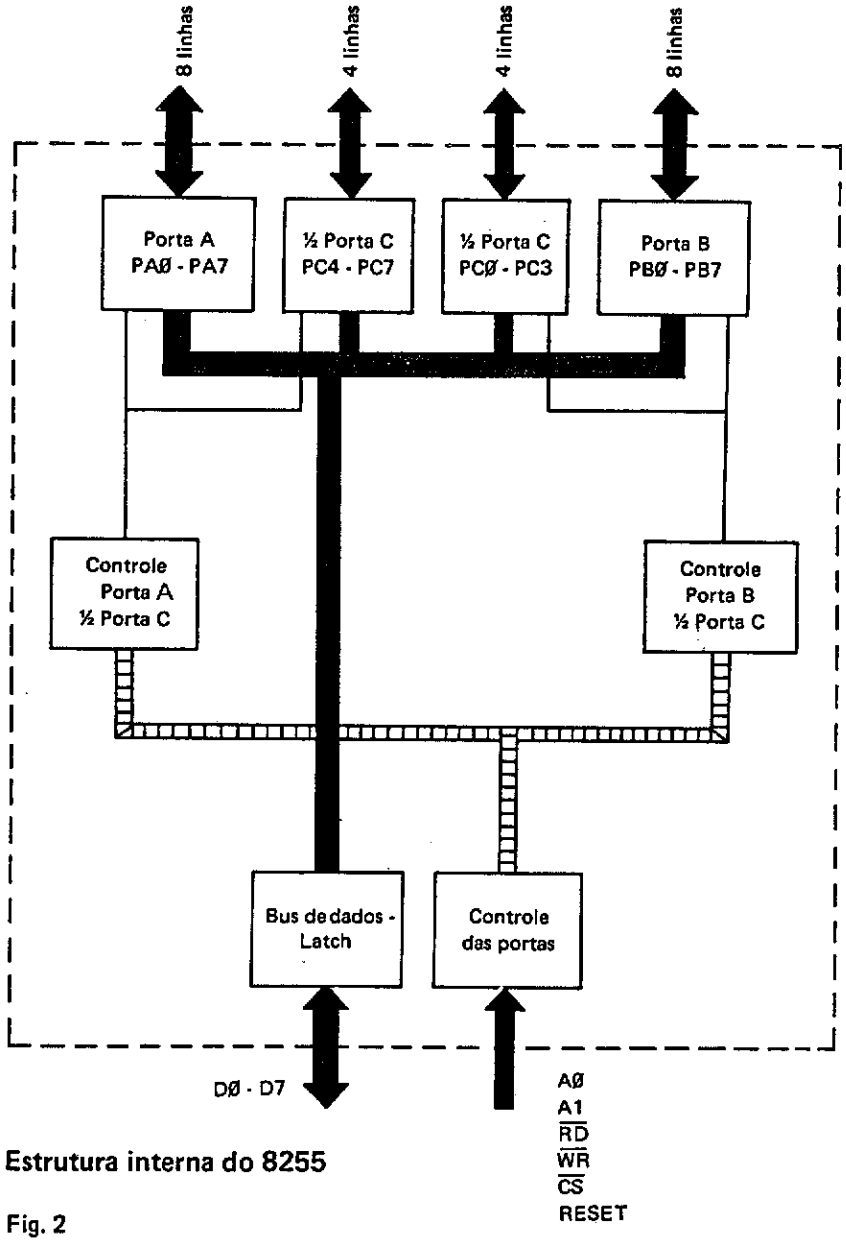


Fig. 1



Estrutura interna do 8255

Fig. 2

A pinologia pode ser conferida na Fig. 3.

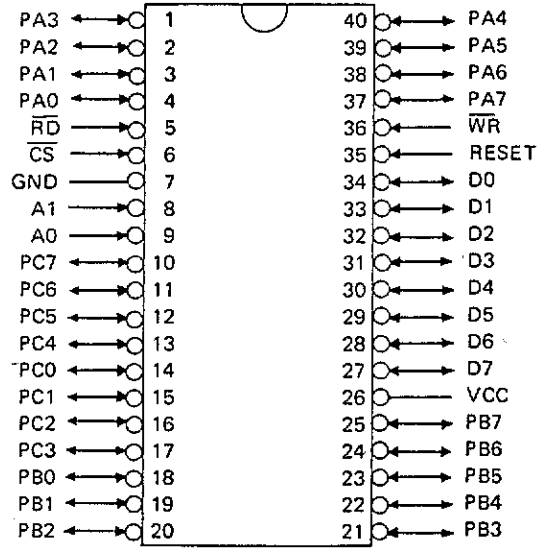


Fig. 3

Pinologia do 8255

Assim, temos as seguintes funções e comandos:

$\overline{RD}$ e $\overline{WR}$ – controlam o vetor de sentido do bus de dados D0 até D7

A0 e A1 – seleção de uma das três portas, ou ainda, a seleção do buffer (para a programação)

$\overline{CS}$ – habilita a comunicação entre o microcomputador e o próprio 8255

RESET – com nível lógico "1", todas as entradas do 8255 vão para o modo Input, isto é, apresentam alta impedância.

VCC – alimentação única de +5 Volts

GND – Terra, negativo

PA0 - 7 – porta A (8 linhas)

PB0 - 7 – porta B (8 linhas)

PC0 - 7 – porta C (8 linhas)

D0 - D7 – bus de dados para o computador

SELEÇÃO DAS PORTAS

Conforme os níveis de AØ e A1, temos as seguintes funções:

AØ	A1	Situação
Ø	Ø	seleção da porta A
1	Ø	seleção da porta B
Ø	1	seleção da porta C
1	1	programação do 8255

A porta A contém um latch/buffer para a saída de dados, e um latch para a entrada de sinais, idêntico à porta B. Já, a porta C está dividida em dois grupos de 4 linhas cada, permitindo a programação diferente para cada grupo. Por exemplo, as linhas PCØ até PC3 podem agir como entradas, enquanto as linhas PC4 até PC7 representam saídas.

MODOS DE SELEÇÃO

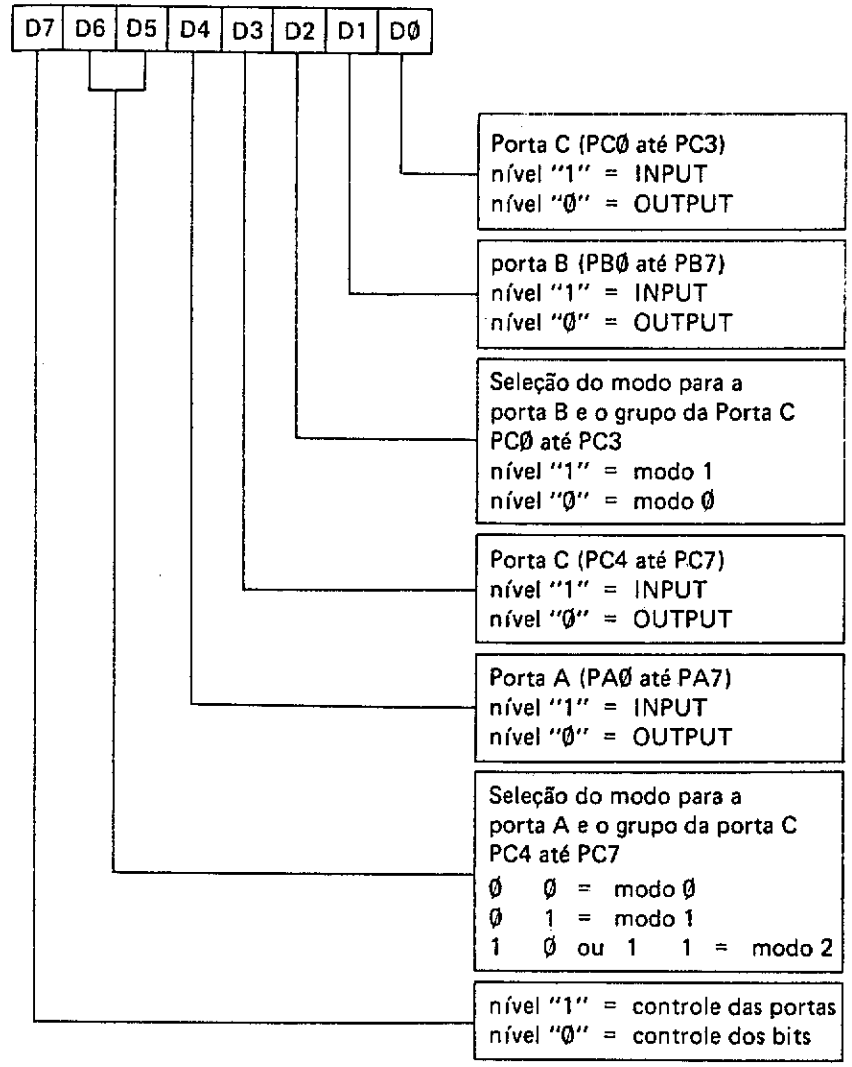
O 8255 apresenta três modos de operação:

Modo Ø = input/output básico
Modo 1 = input/output com strobe
Modo 2 = bus direcional com strobe

A PROGRAMAÇÃO DO 8255 EM SI

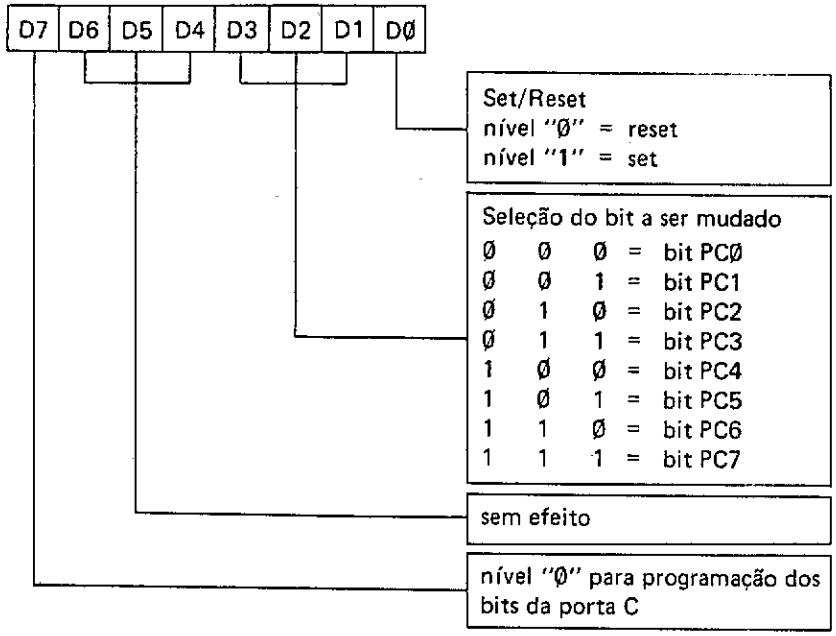
Após ligar a alimentação do 8255, é imprescindível uma palavra de comando, sem a qual este circuito integrado permanece no modo RESET. Esta palavra de comando define o modo e o tipo de funcionamento para cada porta. Lembre-se que para aceitar a programação, as linhas AØ e A1 devem estar em nível "1".

O comando é formado por 8 bits, onde cada bit tem uma função muito especial:



Obs.: O nível de D7 diz ao 8255 de que tipo de palavra de programação trata-se, pois com um nível "1", a palavra será interpretada para controlar as portas, enquanto com D7 em nível "Ø", a palavra representa uma programação para setar os bits da porta C.

Para setar ou "ressetar" os bits da porta C, temos então o seguinte critério na interpretação da palavra de comando:



OS MODOS DE OPERAÇÃO EM DETALHE

1. Modo Ø = Input/Output básico:
   - com este modo pode-se operar com as portas A, B, C usando-as como simples entradas ou saídas, tendo assim:
     - duas portas de 8 bits (A e B)
     - duas portas de 4 bits (C + C)

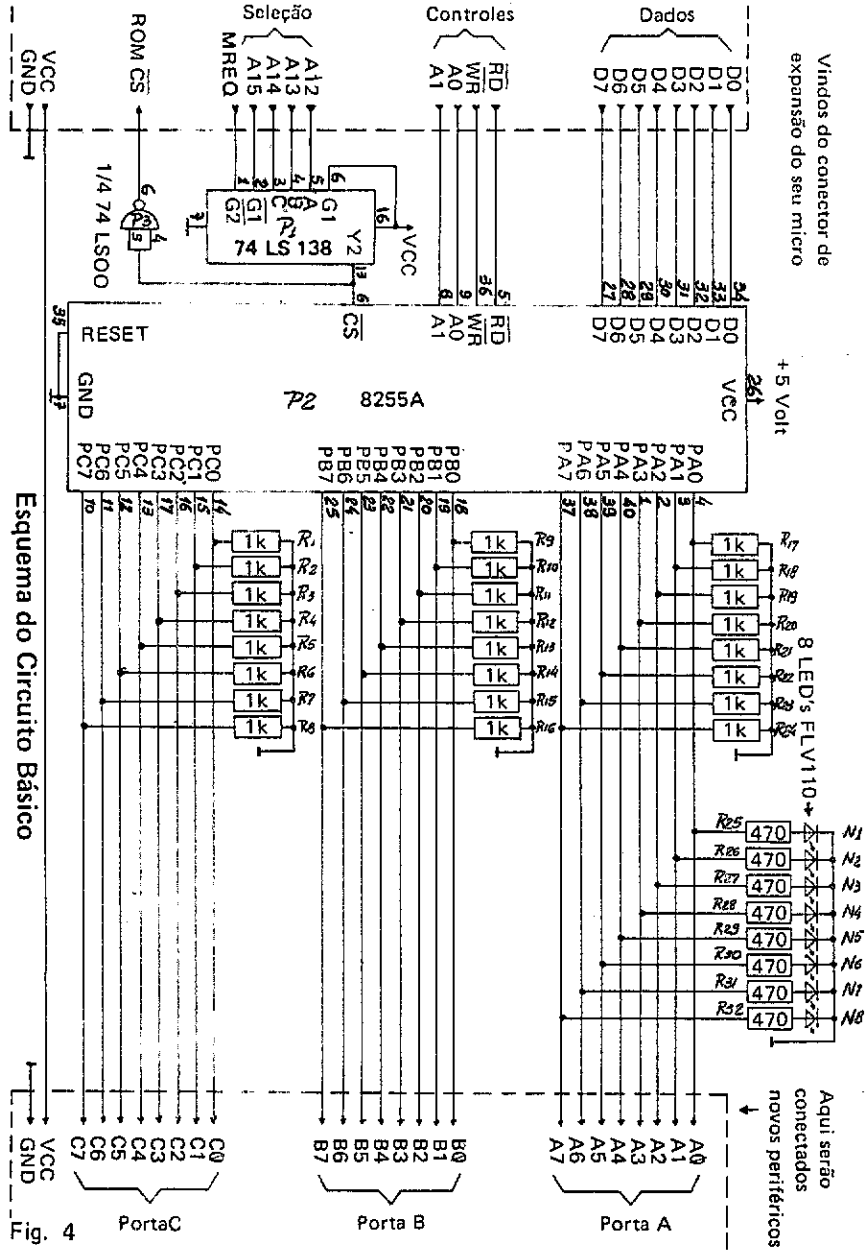
     As saídas têm um latch, isto é, o sinal de saída permanece até que sejam novamente chaveadas.

2. Modo 1 = Input/Output com strobe:
   - neste modo, é feita uma transferência de/ou para uma porta específica, em conjunto com os sinais de strobe e handshaking, onde as portas A e B atuam como as entradas e/ou saídas, enquanto a porta C gera ou verifica os sinais de strobe e handshaking.

3. Modo 2 = bus direcional com strobe:
   - usado para estabelecer uma comunicação entre o computador e um periférico através de até 8 bits em sentido bi-direcional. Neste caso, os 8 bits serão representados pela porta A, enquanto a porta C fornece sinais de controle (5 bits).

Para todos os projetos apresentados, usaremos unicamente o modo 0, deixando um estudo detalhado dos demais modos para literaturas técnicas específicas.

No circuito básico, apresentado no próximo capítulo, usaremos o circuito integrado 8255A, onde o índex "A" representa a velocidade máxima de operação (4,0 MHz), enquanto o tipo normal 8255 sem índex trabalha somente até 2,0 MHz, podendo assim apresentar operações irregulares trabalhando com microcomputadores com uma freqüência de clock elevado (3,25 MHz no caso do TK85).



Esquema do Circuito Básico

Fig. 4

CIRCUITO BÁSICO

Para facilitar a utilização de diversos projetos, economicamente vantajosos, montaremos um circuito básico, a partir do qual, com um mínimo de componentes adicionais, podem ser montados todos os projetos indicados neste livro.

O coração deste circuito básico é o 8255A, que é uma interface de entrada e saída programável. Temos ainda um multiplexador 74LS138, servindo como controlador do 8255A, liberando o circuito somente em determinados endereços. Mais 8 LED's para monitorar sinais e alguns resistores — o suficiente para criar os mais ousados projetos.

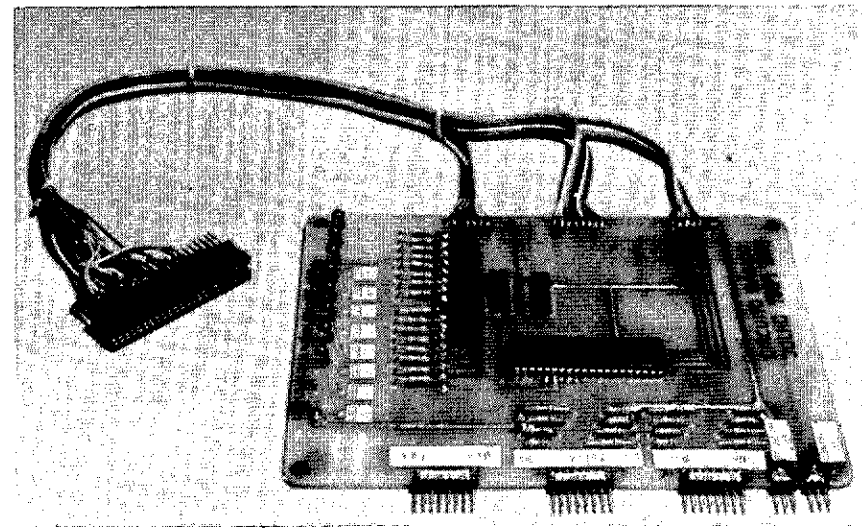


Foto 1

O esquema pode ser visto na Fig. 4, e a montagem em si pode ser feita numa placa de dupla face.

As Figs. 5 + 6 mostram, como deve ser preparada cada face da placa.

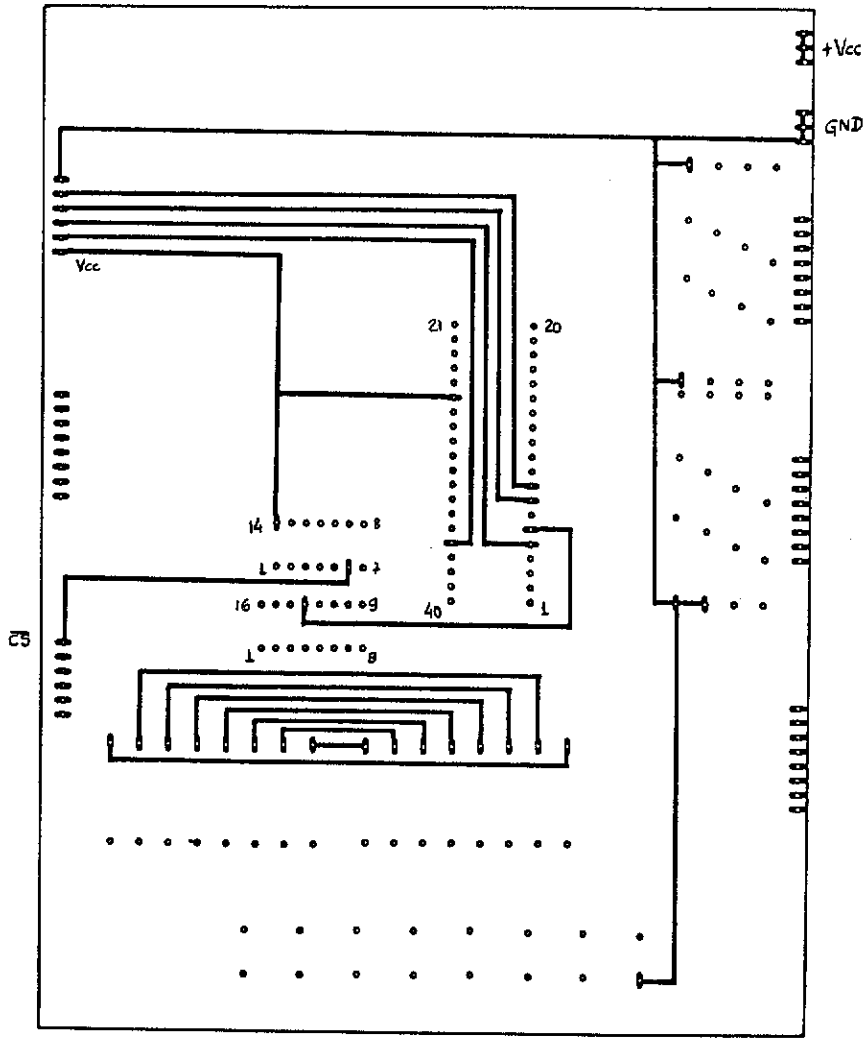


Fig. 5

CIRCUITO BÁSICO
VISTO POR CIMA

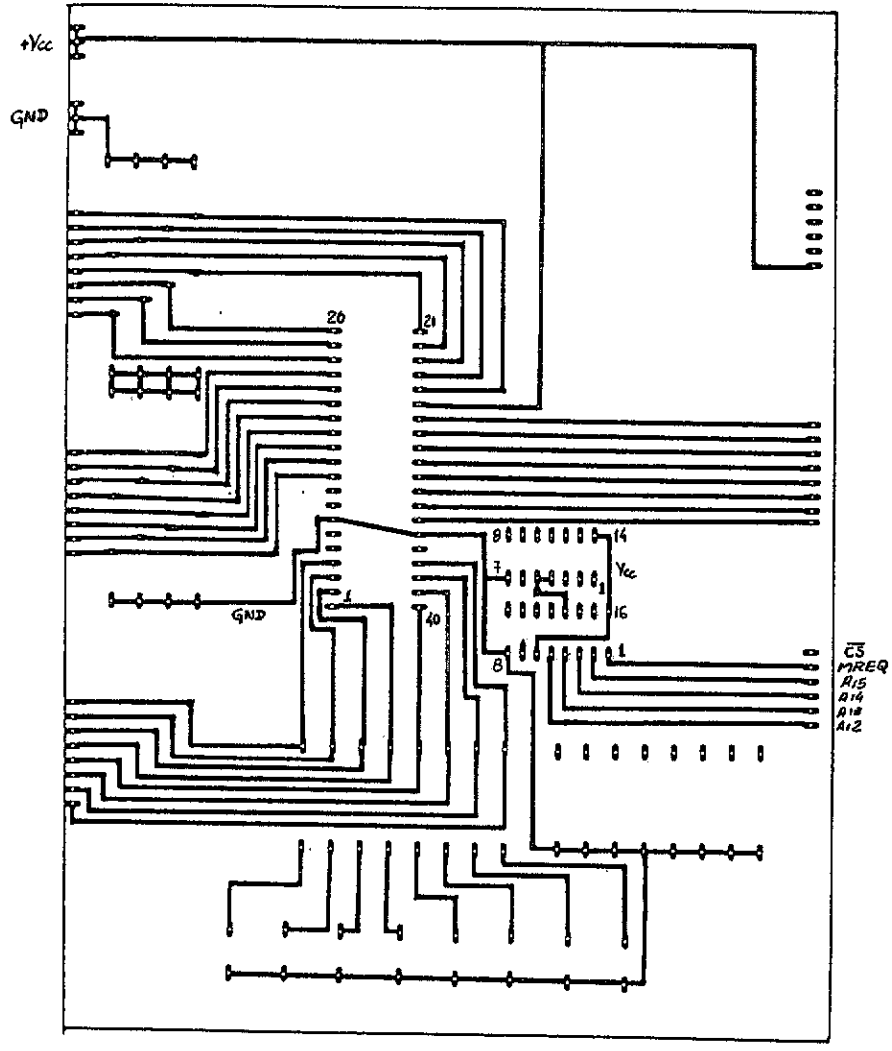
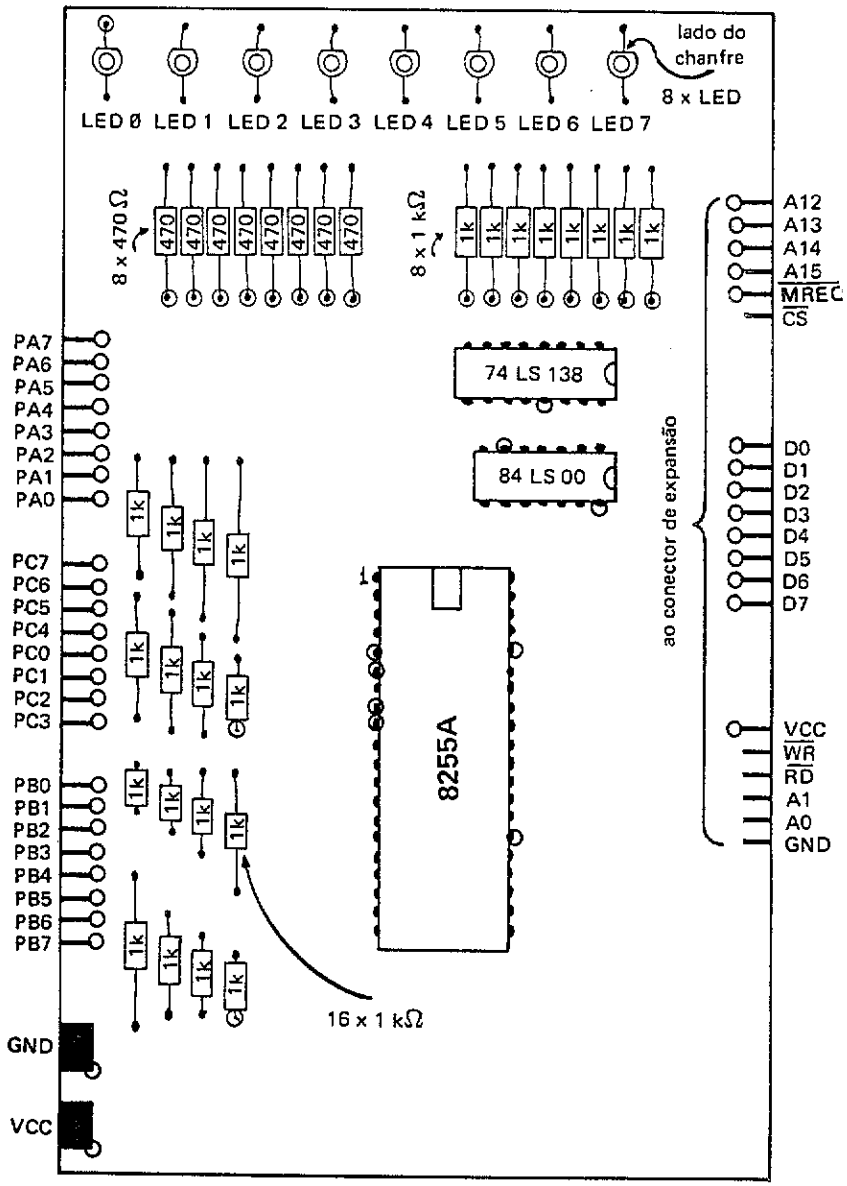


Fig. 6

CIRCUITO BÁSICO
VISTO POR BAIXO



CIRCUITO BÁSICO
O SEU MICRO E O MUNDO EXTERNO

Fig. 7

Relação de material necessário para o circuito básico:

1	circuito integrado 8255A
1	circuito integrado 74LS138
8	LED's FLV 110
8	resistores 470 Ohms 1/8 Watt
24	resistores 1 Kilo-Ohm 1/8 Watt
1	plug para o seu micro (expansão)
1	placa de dupla face conf. Fig. 5 + 6
5m	cabinho
	solda

A MONTAGEM EM SI:

Antes de tudo, prepare o circuito impresso de dupla face de acordo com as Fig. 5 + 6. A seguir, solde os componentes, observando que alguns pinos dos circuitos integrados e alguns resistores devem ser soldados em ambas as partes do circuito impresso. A posição dos componentes é indicada na Fig. 7, e depois da montagem confira com cuidado todo o trabalho feito.

Tendo a placa montada, corte o cabinho em pedaços de no máximo 25 cm de comprimento e solde-os nos pontos indicados na Fig. 8.

Nem todos os projetos apresentados requerem uma memória superior a 2 kBytes (memória interna do seu micro). Assim sendo, você poderá conectar o circuito básico diretamente à expansão do seu micro. Se quiser continuar com a sua expansão de memória (válido para os micros ZX81, TK82, TK83 e NE-Z8000), então poderá montar um "prolongador de expansão" conforme mostra a Fig. 8.

Mesmo para os micros que já vêm com uma memória interna de 16 kBytes ou mais, como é o caso do TK85 e CP200, este prolongador pode ser util. Talvez você tenha uma impressora que queira ligar junto com o circuito básico.

A ligação do plug depende do micro que você tem. Existem atualmente três tipos de ligações:

Tipo 1 — ZX81, TK82, TK83, NE-Z8000, TK85
Tipo 2 — CP200 antigo (sem o High-Speed original)
Tipo 3 — CP200 novo (já com o High-Speed de fábrica)

Para o tipo 1, a ligação deve ser feita de acordo com a Fig. 9.

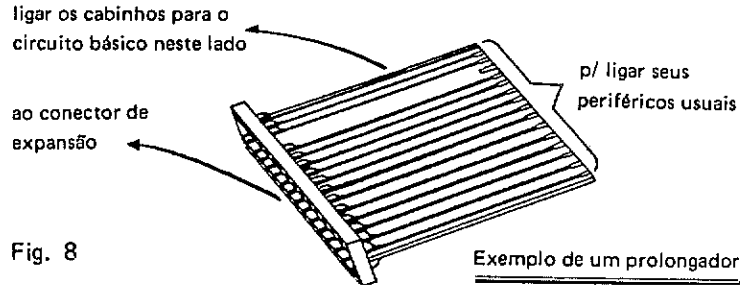


Fig. 8

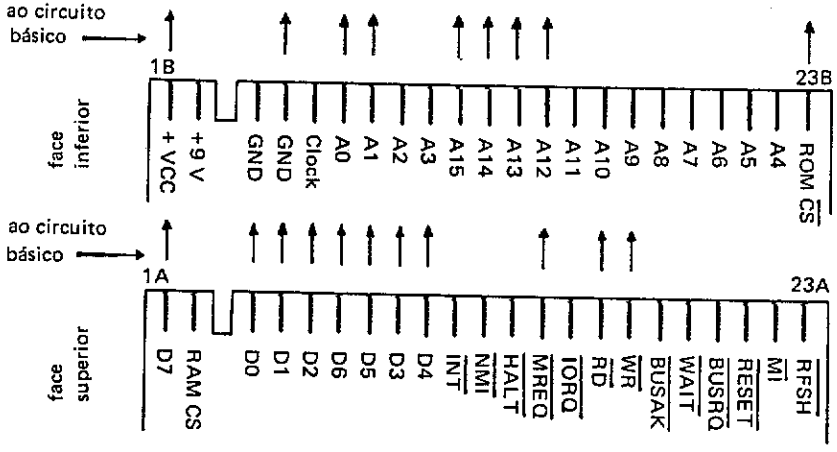


Fig. 9 **ZX81 TK82 TK83 TK85 NE-Z 8000**

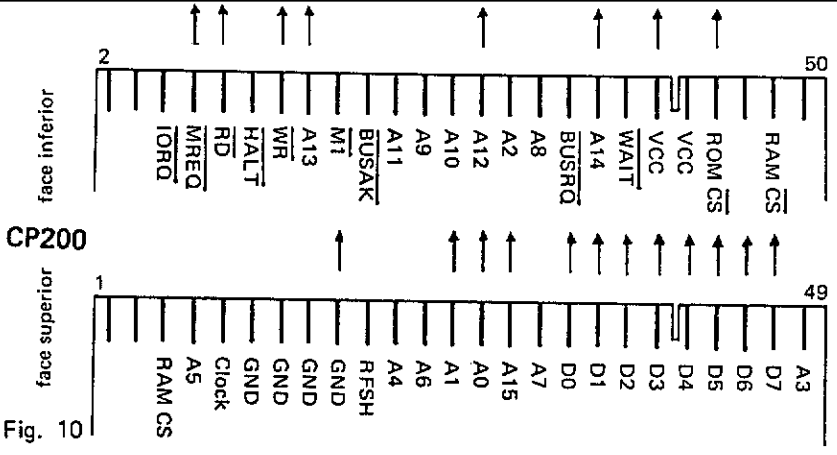


Fig. 10

Quem tem o tipo 2, terá de realizar antes da ligação do plug uma pequena modificação dentro do computador. Não se preocupe, esta modificação não altera o funcionamento normal do seu micro, apenas leva ao condutor de expansão duas linhas ora inexistentes: VCC (+5 Volts) e ROM CS. Na Fig. a seguir pode-se observar, que estas linhas "faltantes" devem ser ligadas na face inferior do conector.

A linha ROM $\overline{CS}$ pode ser produzida usando como base o capítulo "TABELAS E DICAS" na parte do endereçamento interno do micro.

Para os usuários do micro tipo 3, e os do tipo 2 já alterados, prevalece a ligação do plug conforme a Fig. 10.

Tendo tudo ligado, chega a hora de fazer o primeiro teste do circuito básico. Para isto, conecte o plug à expansão do micro e ligue a alimentação.

Se a montagem estiver certa aparecerá, como de costume, o cursor na tela. Se isto acontecer no seu caso, passe para o capítulo seguinte.

O MICRO NÃO FUNCIONA — E AGORA?

Em primeiro lugar, não se apavore. Por mais complicado e por mais frágil que o computador pareça, a probabilidade de "queimar" o micro acidentalmente é mínima. Normalmente, o defeito pode ser facilmente eliminado seguindo as verificações a seguir:

1. Desligue imediatamente o microcomputador.
2. Confira se o plug está encaixado corretamente no micro, e se os contatos deste plug estão perfeitamente alinhados com as pistas do conector de expansão.
3. Verifique se os conectores do plug, onde foram soldados os cabinhos, não estão encostados um ao outro.
4. Seguindo fio por fio, veja se não foi esquecida uma ligação ou se há fios trocados.
5. Confira a posição dos circuitos integrados, observando o chanfre destes componentes, que devem coincidir com a Fig. 7.

6. Observe que alguns dos componentes são soldados em ambas as faces do circuito impresso. Verifique se todas as soldas foram efetuadas.
7. É possível, que durante a confecção do circuito impresso, o ácido não tenha corroído perfeitamente, deixando restos de cobre "curto-circuitando" agora as pistas. Neste caso, use uma faca ou uma chave de fenda para raspar entre as pistas.
8. Usando um multímetro, verifique se existem rupturas nas pistas do circuito impresso. Estas rupturas podem ser microscópicas, imperceptíveis a olho nu. Um pingo de solda resolve este problema.
9. Durante a montagem dos componentes pode ter caído um respingo de solda, provocando agora curto-circuitos. Isto acontece comumente entre os pinos dos circuitos integrados. Use um multímetro para tirar eventuais dúvidas.
10. Um outro defeito muito comum é a rejeição do teclado pelo micro, parcialmente ou totalmente, apesar da existência do cursor na tela. Neste caso, a interligação do circuito básico é comprida demais e deve ser encurtada.
11. Consulte o capítulo "TABELAS E DICAS" na parte "Endereçamento interno do micro".
12. Somente em último caso, troque o 74LS138 e o 8255A.

Seguindo as indicações acima, na maioria dos casos, o problema é resolvido. Todavia, persistindo o defeito, não se esqueça das causas menos comuns, tais como soldas frias, defeito do micro etc...

O PROGRAMA MONITOR

Para introduzir os códigos de máquina dos programas, existem vários métodos. Uma opção é o programa a seguir que introduz os códigos de máquina e apresenta uma soma a cada sete bytes digitados, permitindo então conferir e alterar se necessário. Chamamos este programa de Monitor.

```
 200 FAST
1000 LET A$="00 00 CD 23 0F 01 2
6 00 2A 00 7F E5 09 44 4D 2A 29
40 09 22 29 40 21 0C 40 3E 09 5E
 23 56 D5 EB 09 EB 72 2B 73 23 E
3 3D 28 03 D1 18 EE E1 E5 01 7C
40 A7 ED 42 44 4D E1 ED B8 21 7D
 40 36 00 23 36 00 23 C1 03 03
1 23 70 23 36 EA 0B 0B 23 11 01
00 EB 19 EB 36 3D ED B0 36 76 34
 CD 2B 0F C9"
1050 LET NRTP=(PEEK 16388+256*PE
EK 16389)-256
1100 FOR I=1 TO LEN A$ STEP 3
1200 POKE (NRTP+INT (I/3)),16*CO
DE A$(I)+CODE A$(I+1)-476
1300 NEXT I
1350 SLOW
1400 PRINT "Nº DE BYTES ?"
1500 INPUT N
1600 POKE NRTP,N-256*INT (N/256)
1700 POKE (NRTP+1),INT (N/256)
1800 RAND USR (NRTP+2)
1900 CLS
2100 LET I=16514
2300 LET F=16513+N
2800 FOR M=I TO F STEP 7
2900 PRINT AT 21,0;"------------
--------------------"
3000 SCROLL
3100 LET T=0
3200 DIM A$(32)
3300 LET A$(1 TO 5)=STR$ M
```

```
3400 PRINT AT 21,0 M;" ";
3500 FOR J=1 TO 21 STEP 3
3600 IF (M+INT (J/3))/2<>INT ((M+INT (J/3))/2) THEN GOT
O 5000
3700 FOR K=0 TO 1
3800 IF INKEY$<>"" THEN GOTO 380
0
3900 IF INKEY$="" THEN GOTO 3900
4000 LET P=CODE INKEY$
4100 IF P<28 OR P>43 THEN GOTO 3
800
4200 LET A$(J+K+6)=CHR$ P
4300 PRINT CHR$ P;
4400 NEXT K
4500 LET N=16*CODE A$(J+6)+CODE
A$(J+7)-476
4600 LET T=T+N
4700 POKE (M+INT (J/3)),N
4800 PRINT " ";
4900 NEXT J
5000 PRINT TAB 27;T
5100 LET A$(28 TO )=STR$ T
5200 IF INKEY$="S" THEN GOTO 560
0
5300 IF INKEY$<>"N" THEN GOTO 52
00
5500 GOTO 3100
5600 PRINT AT 20,0;A$
5610 PRINT AT 21,0;"
                    "
5700 NEXT M
```

Digite o programa e a seguir comande RUN e NEW LINE para rodá-lo.

Responda a pergunta "N– de bytes?" com o valor de N, apresentado junto a cada rotina assembly. O programa criará então uma linha REM que armazenará todos os bytes da rotina.

A seguir, aparecerá na tela o endereço 16514 e você deverá digitar os códigos correspondentes ao endereço apresentado. Ao terminar cada linha, o programa fornecerá a soma em decimal dos códigos digitados. Se este valor não coincidir com o do livro, pressione a tecla N e redigite os códigos dessa linha. Caso contrário digite S e aparecerá outro endereço. Continue a digitar, sempre verificando a soma com o valor no livro.

Para ilustração, vai aqui um exemplo:

```
ROTINA ASSEMBLY 1  N=7
#16514 2A 0C 40 23 36 80 C9 536
```


VERIFICAÇÕES FINAIS

O objetivo deste capítulo é testar se o circuito básico está aceitando os comandos do micro e se todos os LED's acendem. Mesmo com o computador funcionando perfeitamente, ainda há possibilidades de erros de montagem ou de componentes defeituosos.

Dê agora os seguintes comandos diretos e observe o circuito básico:

POKE 8195,128
POKE 8192,255

Todos os LED's devem acender. A luminosidade destes componentes é controlada pelos resistores de 470 Ohms. Reduzindo este valor para 330 Ohms, a luminosidade aumenta, mas em compensação a vida útil dos LED's vê-se reduzida.

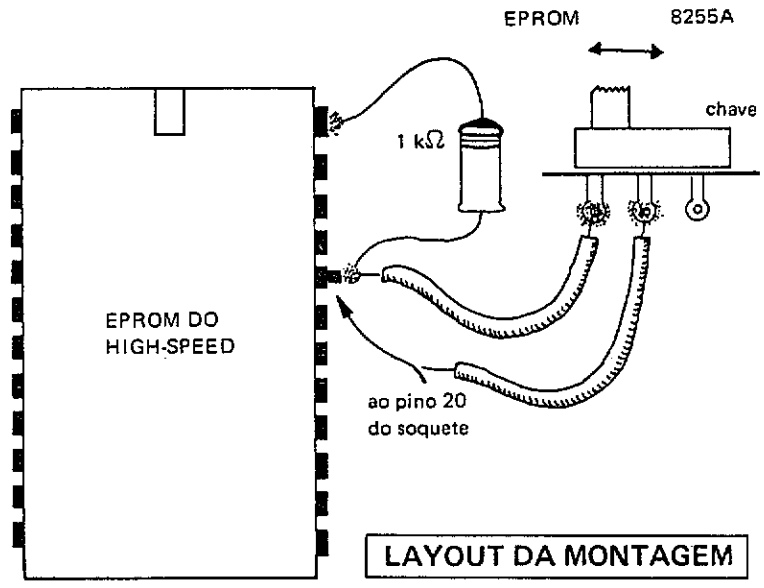
Se um ou mais LED's permanecem apagados, então confira a posição do chanfre destes componentes (Fig. 7), o valor do resistor de 470 Ohms (amarelo – violeta – marrom), e no último caso substitua o LED.

Uma vez com todos os LED's acesos, chegamos a um ponto crucial, para a tristeza de muitos usuários. Digite

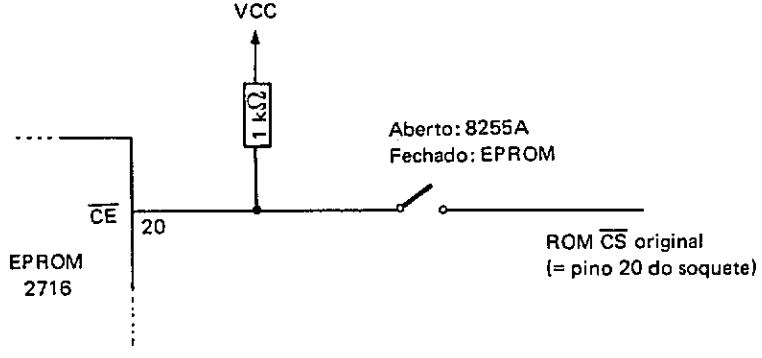
POKE 8192,Ø

e todos os LED's devem apagar.

Se no seu circuito básico todos os LED's apagam, então passe para o capítulo seguinte.



LAYOUT DA MONTAGEM



ESQUEMA

Fig. 11

OS LED'S NÃO APAGARAM

Normalmente, isto não pode ser considerado como um defeito do circuito básico, mas sim um problema de endereçamento dentro do seu microcomputador. Geralmente, isto acontece nos micros TK85, CP200 e nas outras marcas, onde foram incluídas as funções do High-Speed posteriormente.

Nestes casos, os endereços de 8192 até 8195 são ocupados pelo "eco" da EPROM, que contém o High-Speed, e pelo conector de expansão não é possível eliminar este eco. Mas com uma pequena incrementação podemos também sanar este problema.

Se o seu micro for um TK85, então há a alternativa de retirar a EPROM do High-Speed do seu soquete, cada vez que quiser trabalhar com o circuito básico. Esta EPROM é um circuito integrado de 24 pinos, localizado à esquerda de um circuito integrado (ou dois) igual. Este procedimento é delicado, pois o perigo de entortar as pernas desta EPROM durante a colocação e a retirada deste componente do micro é muito grande, além de eliminar a possibilidade de usar o circuito básico e o High-Speed ao mesmo tempo.

Você pode então optar pela seguinte montagem, válida para todos os microcomputadores com este tipo de problema.

Devemos ligar uma chave no computador, que deve ser acionada toda vez que se usa o circuito básico.

Podemos ver na figura acima, que o pino nº 20 da EPROM do High-Speed foi isolado do circuito do micro, e a chave, ligada entre este pino e a posição anterior do mesmo.

Agora, cada vez, que se quiser usar o High-Speed, a posição da chave é "EPROM" enquanto que durante o uso do circuito básico, a posição da chave é "8255A". Esta seleção pode ser feita mesmo com o microcomputador ligado.

POR QUE ACENDEM OS LED'S?

Vimos no capítulo anterior, que um simples POKE pode comandar os LED's do circuito básico. Vale a pena estudar este capítulo, pois um maior conhecimento irá ajudá-lo a criar novos projetos.

O POKE

Cada comando POKE exige dois números separados por uma vírgula. O primeiro número representa o endereço da memória, que varia no nosso caso entre 8192 e 8195. O número depois da vírgula é o valor que deve ser colocado nestes endereços. Este dado está limitado entre Ø e 255, pois o computador só pode manipular 8 bits por posição de memória. Mas, afinal o que é um bit?

O BIT

Um bit representa a base matemática fundamental do microcomputador, e indica sempre um dos dois estados possíveis:

bit Ø = negativo, sem sinal
bit 1 = positivo, com sinal

Sendo assim, com apenas 1 bit podemos representar dois números, o Ø e o 1. Para uma terceira combinação é necessário mais um bit:

bit 1	bit Ø		
Ø	Ø	=	valor Ø
Ø	1	=	valor 1
1	Ø	=	valor 2

e, completando as combinações possíveis:

1 1 = valor 3

Como podemos ver, temos quatro combinações possíveis com dois bits, e continuando com 3 bits temos então 8 configurações diferentes:

bit 2	bit 1	bit Ø		
Ø	Ø	Ø	=	valor Ø
Ø	Ø	1	=	valor 1
Ø	1	Ø	=	valor 2
Ø	1	1	=	valor 3
1	Ø	Ø	=	valor 4
1	Ø	1	=	valor 5
1	1	Ø	=	valor 6
1	1	1	=	valor 7

Para calcular a quantidade das combinações possíveis a partir do número de bits disponível, podemos utilizar a seguinte fórmula:

$$nº\ combinações = 2^{nº\ bits}$$

Sabemos que o nosso micro manipula 8 bits por posição de memória. Assim sendo, a quantidade de combinações possíveis é

$$2^8 = 256$$

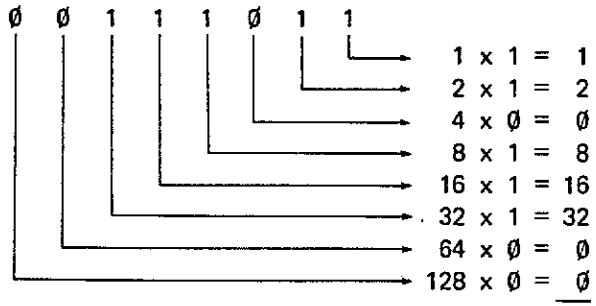
Considerando o próprio Ø como uma das combinações, chegamos assim ao número 255 como valor máximo por endereço.

Os oito bits do computador são numerados de D0 até D7, onde cada bit representa um valor específico de acordo com a sua posição dentro do byte (byte = conjunto de 8 bits).

valor do bit:	128	64	32	16	8	4	2	1
número do bit:	7	6	5	4	3	2	1	Ø

Somando todos os valores acima, chegamos novamente ao valor 255.

Desta maneira, se o formato de um byte é Ø Ø 1 1 1 Ø 1 1, podemos calcular o valor decimal correspondente da seguinte maneira:



Como pode-se ver, somente serão somados os valores dos bits que estão com nível 1. Os demais serão ignorados.

Também podemos usar uma fórmula matemática para chegar ao valor decimal, já que sabemos que os bits Ø, 1, 3, 4 e 5 estão com nível 1:

$$2^{\emptyset} + 2^{1} + 2^{3} + 2^{4} + 2^{5} = 59$$

O circuito básico indica com os LED's o estado de cada um dos 8 bits. Quando um bit for em nível 1, o respectivo LED acenderá, permanecendo apagado quando o bit estiver em nível Ø.

Desta maneira, se quisermos acender apenas os LED's Ø, 2 e 6, teremos que escrever primeiramente a configuração binária correspondente:

Ø 1 Ø Ø Ø 1 Ø 1

Assim, o valor decimal correto será 69, como pode ser visto na Fig. 12.

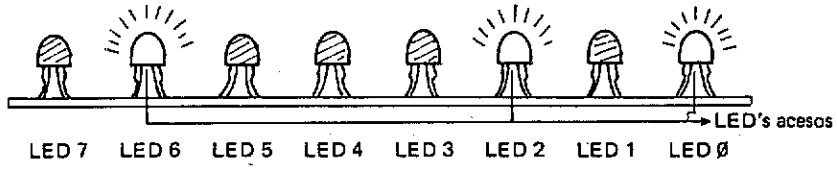


Fig. 12

Mais informações sobre a programação do circuito básico podem ser encontradas no final deste livro.

LUZES SEQÜENCIAIS

A partir de agora, todos os projetos apresentados são feitos para um funcionamento na modalidade SLOW do microcomputador. Todavia, nada impede que usemos as mesmas rotinas com o micro trabalhando em FAST. Devemos, porém, ter em mente que todo o funcionamento se processará muito mais rapidamente, requerendo eventuais atrasos adicionais (loops FOR/NEXT).

Vamos então à nossa primeira rotina:

ROTINA BASIC 2

```
10 POKE 8195,128
20 FOR F=Ø TO 7
30 POKE 8192,2↑F
40 NEXT F
50 GOTO 20
```

Rodando o programa acima, um LED após o outro irá acender e apagar, começando pelo LED Ø até o LED 7, repetindo esta seqüência até pararmos o programa com BREAK.

A velocidade não é das maiores, e rodando a mesma rotina em FAST, o visual já melhora, mas continua bastante lento.

Podemos substituir o comando POKE por uma pequena rotina em linguagem de máquina, como demonstra o programa a seguir:

ROTINA ASSEMBLY 2 N=11

```
*16514 3E 80 32 03 20 3E 00 337
 16521 32 00 20 C9           283
```

ROTINA BASIC 3

```
10 FOR F=0 TO 7
20 POKE 16520,2**F
30 RAND USR 16514
40 NEXT F
50 RUN
```

Por enquanto, a velocidade deste programa em comparação com o programa totalmente em BASIC permaneceu inalterada. O objetivo é apenas mostrar, que podem ser intercaladas rotinas de ASSEMBLY no programa BASIC para criar os mesmos efeitos. Mas, qual é a vantagem?

Bem, escrevendo todo o programa BASIC em linguagem de máquina, teremos acesso a velocidades muito superiores à percepção dos nossos olhos, conseguindo assim qualquer velocidade conforme o desejo de cada um.

Na prática, uma rotina "pura" em linguagem de máquina seria por exemplo esta:

ROTINA ASSEMBLY 3 N=46

```
*16514 18 0B 06 0F C5 06 00 259
 16521 10 FE C1 10 F8 C9 3E 990
*16528 80 32 03 20 06 0A C5 426
 16535 3E 01 06 08 C5 32 00 324
*16542 20 CD 84 40 17 C1 A7 816
 16549 10 F4 C1 10 EC 3E 00 767
*16556 32 00 20 C9             283
```

A única instrução necessária para rodar este programa, é o comando direto RAND USR 16514.

Após dar o comando acima, a luz seqüencial reinicia-se até completar dez seqüências completas, após as quais o programa retorna ao BASIC.

Percebemos que a velocidade aumentou consideravelmente, e talvez haja até necessidade de mudar esta velocidade. Para isto, podemos adicionar à nossa rotina ASSEMBLY o seguinte programa BASIC auxiliar:

ROTINA BASIC 4

```
 0 REM   (ROTINA ASSEMBL. 3
10 PRINT "VELOCIDADE (1-255) =
20 INPUT V
30 PRINT V
40 POKE 16517,V
50 RAND USR 16514
60 RUN
```

O valor original de velocidade é 15, e números maiores que 50 provocam uma demora muito grande até que o programa volte ao BASIC. Para reduzir ou aumentar a quantidade dos loops completos, basta um POKE no endereço 16553 (valor original = 10), que define a quantidade das seqüências completas por vez. A única exceção é o valor Ø, pois ao contrário do que poderíamos pensar, ou seja, que não haveria nenhum loop com um POKE 16553,Ø serão realizados 256 loops antes de retornar-se ao BASIC.

Se quisermos deixar funcionar o seqüencial o tempo todo, podemos utilizar o programa BASIC a seguir:

ROTINA BASIC 5

```
 0 REM   (ROTINA ASSEMBLY 3)
10 RAND USR 16514
20 RUN
```

ou, em linguagem de máquina, temos respectivamente:

ROTINA ASSEMBLY 4 N=47

```
*16514 18 0B 06 0F C5 06 00 259
 16521 10 FE C1 10 F8 C9 3E 990
*16528 80 32 03 20 06 0A C5 426
 16535 3E 01 06 08 C5 32 00 324
*16542 20 CD 84 40 17 C1 A7 816
 16549 10 F4 C1 10 EC 3E 00 767
*16556 32 00 20 18 E3          333
```

Digite RAND USR 16514 para iniciar. Observe, que agora a única maneira de parar o programa é desligar o microcomputador.

Substituindo os LED's por relés ou tiristores, podemos chavear lâmpadas de potência e tensões maiores, chamando por exemplo à atenção para a sua vitrine, criando efeitos visuais numa discoteca ou outros inúmeros projetos, limitados apenas pela sua criatividade.

OUTROS EFEITOS COM OS LED'S

Há várias maneiras de produzir efeitos de luz com os LED's, que não se restringem apenas a luzes seqüenciais. As sugestões a seguir sempre trazem um programa escrito em BASIC e uma rotina correspondente em linguagem de máquina. Vale lembrar, que todas as rotinas ASSEMBLY contém atrasos, calculados para um funcionamento na modalidade SLOW do computador. Caso queira trabalhar em FAST (por exemplo devido à inexistência do SLOW no seu micro), então estes atrasos devem ser "re-calibrados". Isto é feito com um POKE nos endereços indicados, onde o atraso aumenta proporcionalmente com o valor do número n (POKE (endereço), n).

O primeiro programa deixa "correr" um LED da esquerda para à direita e vice-versa.

ROTINA BASIC 5

```
10 POKE 8195,126
20 FOR F=0 TO 7
30 POKE 8192,2**F
40 NEXT F
50 FOR F=6 TO 1 STEP -1
60 POKE 8192,2**F
70 NEXT F
80 GOTO 20
```

Ou ainda, em linguagem de máquina:

ROTINA ASSEMBLY 5 N=59

```
+16514 18 0B 06 0F C5 06 00 259
 16521 10 FE C1 10 F8 C9 3E 990
+16528 80 32 03 20 06 0A C5 426
 16535 3E 01 06 08 C5 32 00 324
+16542 20 CD 84 40 17 C1 10 665
 16549 F5 06 08 C5 1F 32 00 537
+16556 20 CD 84 40 C1 A7 10 809
 16563 F4 C1 10 DF 3E 00 32 788
+16570 00 20 C9                 233
```

Inicie a rotina acima com RAND USR 16514.

O endereço do atraso é 16517, e usando um programa BASIC auxiliar, semelhante ao capítulo anterior, podemos mudar esta velocidade:

ROTINA BASIC 7

```
 0 REM   (ROTINA ASSEMBLY 5.
10 PRINT "VELOCIDADE (1-255) ="

20 INPUT V
30 PRINT V
40 POKE 16517,V
50 RAND USR 16514
60 GOTO 10
```

Também aqui temos uma seqüência de dez voltas completas, antes da rotina ASSEMBLY rotornar ao BASIC. Como no capítulo das luzes seqüenciais, esta quantidade pode ser alterada mediante um POKE no endereço 16533.

Uma outra maneira é um tipo de luz seqüencial, onde acendem sempre dois LED's ao mesmo tempo, com três LED's apagados entre eles:

ROTINA BASIC 8

```
 10 POKE 8195,128
 20 POKE 8192,17
 30 GOSUB 110
 40 POKE 8192,34
 50 GOSUB 110
 60 POKE 8192,68
 70 GOSUB 110
 80 POKE 8192,136
 90 GOSUB 110
100 GOTO 20
110 FOR F=1 TO 2
120 NEXT F
130 RETURN
```


E em linguagem de máquina:

ROTINA ASSEMBLY 8 N=82

```
*16514 18 0B 06 0F C5 06 00 259
 16521 10 FE C1 10 F6 C9 3E 990
*16528 80 32 03 20 06 0A C5 426
 16535 3E 11 32 00 20 CD 84 498
*16542 40 3E 22 32 00 20 CD 447
 16549 84 40 3E 44 32 00 20 408
*16556 CD 84 40 3E 88 32 00 649
 16563 20 CD 84 40 C1 10 DC 862
*16570 3E 00 32 00 20 C9    345
```

Nesta rotina, a velocidade é também controlada no endereço 16517, a quantidade das voltas completas no endereço 16533. Para iniciar, digite RAND USR 16514.

E, por fim, mais um efeito simples, mas bonito:

ROTINA BASIC 9

```
 10 POKE 8195,128
 20 LET A=0
 30 POKE 8192,A
 40 FOR F=0 TO 7
 50 LET B=A
 60 LET A=A+2**F
 70 FOR G=1 TO 5
 80 GOSUB 210
 90 POKE 8192,B
100 GOSUB 210
110 POKE 8192,A
120 NEXT G
130 NEXT F
140 FOR F=1 TO 10
150 GOSUB 210
160 POKE 8192,255
170 GOSUB 210
180 POKE 8192,0
190 NEXT F
200 GOTO 20
210 RETURN
```

```
ROTINA ASSEMBLY 7    N=133

*16514  16 26 3E 0F 35 06 00 286
 16521  10 FE 11 10 F8 C9 00 920
*16528  00 06 05 C5 3A 8F 40 473
 16535  32 00 20 CD 84 40 3A 541
*16542  90 40 3C 00 20 CD 84 627
 16549  40 C1 10 EF C9 3E 00 800
*16556  32 00 20 06 0A C5 21 331
 16563  01 00 22 8F 40 CD 91 590
*16570  40 21 03 01 22 8F 40 348
 16577  CD 91 40 21 07 03 22 491
*16584  8F 40 CD 91 40 21 0F 669
 16591  07 22 8F 40 CD 91 40 662
*16598  21 1F 0F 22 8F 40 CD 525
 16605  91 40 21 3F 1F 22 8F 513
*16612  40 CD 91 40 21 7F 3F 701
 16619  22 8F 40 CD 91 40 21 688
*16626  FF 7F 22 8F 40 CD 91 973
 16633  40 21 FF 00 22 8F 40 593
*16640  CD 91 40 C1 10 AB C9 995
```

```
VELOCIDADE:16517
QUANTIDADE DE LOOPS:16560
INICIO:RAND USR 16514
```


USANDO AS ENTRADAS DO CIRCUITO BÁSICO

O nosso circuito básico não só pode fornecer sinais de saída, como por exemplo, para o chaveamento dos LED's, como também pode receber sinais externos e transmiti-los ao computador.

Para testar esta modalidade em todas as 24 linhas, escreva o programa a seguir:

```
ROTINA BASIC 10

  10 POKE 8195,155
  20 PRINT AT 9 0;"PORTA A = ";P
EEK 8192;"  "
  30 PRINT AT 11,0;"PORTA B = ";
PEEK 8193;"  "
  40 PRINT AT 13,0;"PORTA C = ";
PEEK 8194;"  "
  50 GOTO 20
```

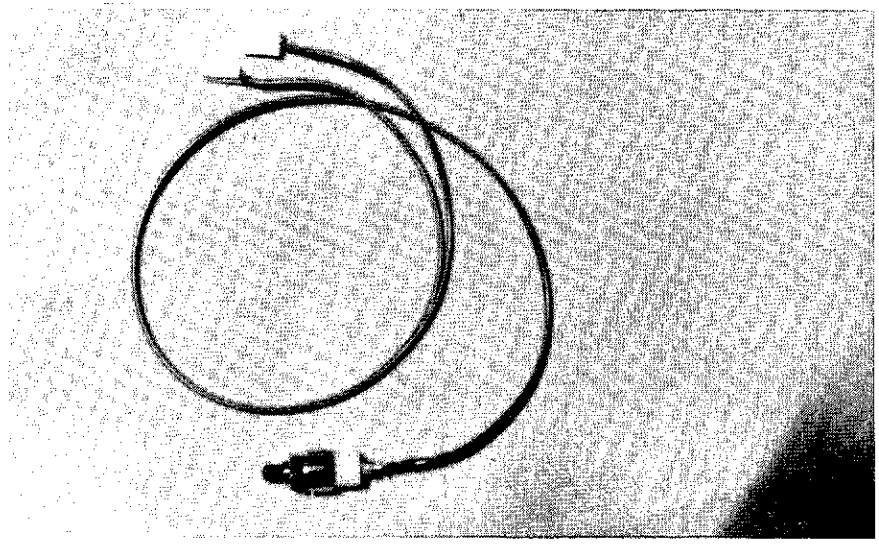


Foto 2

Com um pedaço de fio, ligado numa extremidade no VCC (+5 Volts), encoste nos pontos indicados na tabela abaixo e confira a leitura do micro com os valores indicados:

Entrada "1" (VCC)	leitura:
PA Ø	Port A = 1
PA 1	Port A = 2
PA 2	Port A = 4
PA 3	Port A = 8
PA 4	Port A = 16
PA 5	Port A = 32
PA 6	Port A = 64
PA 7	Port A = 128
tudo livre	Port A = Ø

Devem ser semelhantes as leituras com os Ports B e C. Se porventura houver uma seqüência trocada, houve então uma troca na denominação dos pontos no seu circuito básico. Por outro lado, se uma das entradas não reagir, deve ser inspecionado o circuito impresso, as pontas de solda, e em último caso ser trocado o 8255A. Todavia, na maioria dos casos, mesmo com uma ou mais entradas do 8255A defeituosas, o circuito pode ser usado, trabalhando simplesmente com os demais ports ou entradas.

Uma vez testada também esta modalidade do 8255A, dificilmente haverá problemas nos projetos seguintes, garantindo assim sucesso total nas suas montagens.

TIRO AO ALVO

Este projeto simples, demonstra como o circuito básico pode trabalhar usando entradas e saídas ao mesmo tempo, como por exemplo neste famoso joguinho "Tiro ao Alvo".

Para quem não conhece as regras do jogo, vão aqui as instruções:

toda vez, quando um determinado LED (ou lâmpada) acender, deve-se apertar imediatamente o botão de disparo, "atirando" desta maneira no LED aceso. Se acertar, ganhará um ponto. Por outro lado, se atirar cedo ou tarde demais, perderá um ponto.

O programa a seguir foi escrito para o jogador ter 10 tiros, com uma contagem inicial de 5 pontos. Assim sendo, temos como contagem máxima 15 pontos, e num fracasso total apenas 5 pontos.

O primeiro exemplo, escrito em BASIC, serve apenas para demonstrar o funcionamento, pois ainda é bastante fácil acertar o LED 3, que escolhemos como o "alvo" do nosso jogo. O botão de disparo deve ser ligado na linha PCØ.

ROTINA BASIC 11

```
10 POKE 8195,137
20 LET T=0
30 LET P=5
40 FOR F=0 TO 7
50 POKE 8192,2**F
60 IF PEEK 8194<>0 THEN GOTO 9
0
70 NEXT F
80 GOTO 40
```

```
  90 IF F<>3 THEN GOTO 150
 100 LET P=P+1
 110 LET T=T+1
 120 IF PEEK 8194  0 THEN GOTO
20
 130 IF T=11 THEN GOTO 170
 140 GOTO 40
 150 LET P=P-1
 160 GOTO 110
 170 PRINT "VOCE FEZ ";P;" PONTO
S."
```

Todos fizeram pelo menos 10 pontos? Ótimo, então podemos passar para um desafio maior, usando desta vez linguagem de máquina:

ROTINA ASSEMBLY 8   N=124

*16514	00	00	C5	06	0A	C5	06	416
16521	00	10	FE	C1	10	F8	C1	920
*16528	C9	3E	89	32	03	20	3E	547
16535	00	32	82	40	32	83	40	489
*16542	05	07	3E	01	32	00	20	158
16549	CD	84	40	F5	3A	02	20	738
*16556	FE	00	20	06	F1	17	10	572
16563	EE	18	E8	F1	A7	FE	08	1164
*16570	20	07	3A	83	40	3C	32	402
16577	83	40	3A	82	40	3C	32	557
*16584	82	40	A7	FE	0A	28	09	674
16591	3A	02	20	FE	00	20	F9	627
*16598	18	C6	01	08	0B	CD	F5	692
16605	08	01	10	00	11	EE	40	344
*16612	CD	6B	0B	3A	83	40	C6	774
16619	1C	D7	C9	36	39	29	1B	623
*16626	29	2A	00	26	28	2A	37	258
16633	39	34	38	0E	00			179

Se achar a velocidade muito alta, poderá reduzi-la com um POKE no endereço 16518 (valor inicial = 10). A rotina acima é iniciada com RAND USR 16529, e a quantidade de acertos será indicada na tela após o 10º tiro.

O projeto a seguir, já em estilo profissional, aumenta a velocidade dos LED's à medida que aumenta a contagem do jogador, além de indicar simultaneamente na tela a quantidade de tiros gastos.

ROTINA ASSEMBL 5   N=205

*16514	00	00	[illegible]	06	[illegible]	[illegible]	06	[illegible]
16521	30	10	FE	[illegible]	10	F8	C1	1046
*16528	C9	00	3E	3C	29	18	00	380
16535	39	2E	37	34	38	0E	00	280
*16542	36	39	2C	1E	00	26	28	257
16549	2A	37	3C	34	36	0E	00	276
*16556	3E	2C	32	3E	40	3E	89	544
16563	32	03	20	3E	00	32	82	327
*16570	40	32	33	40	3E	01	32	422
16577	91	40	05	0E	C5	01	09	430
*16584	0A	CD	F5	06	21	0C	00	481
16591	11	92	40	CD	6B	0B	3A	608
*16598	82	40	FE	0A	20	07	3E	559
16605	1D	D7	3E	0C	00	00	C6	504
*16612	1C	D7	01	09	0C	CD	F5	715
16619	08	01	0E	00	11	9E	40	262
*16626	CD	6B	0B	3A	33	40	FE	830
16633	0A	20	07	3E	1D	D7	3E	417
*16640	00	00	00	C6	1C	D7	3A	499
16647	91	40	32	00	20	17	32	364
*16654	91	40	CD	84	40	3A	02	670
16661	20	FE	00	20	05	C1	10	532
*16668	A8	18	9F	C1	3A	91	40	811
16675	FE	10	28	16	3A	82	40	584
*16682	3C	32	82	40	FE	0B	C8	769
16689	3A	02	20	FE	00	20	F9	627
*16696	CD	84	40	16	E0	3A	83	836
16703	40	3C	32	83	40	3A	86	561
*16710	40	A7	D6	03	32	86	40	596
16717	18	D8						240

A velocidade inicial depende do valor no endereço 16521 (normal = 128 e o início do programa é obtido com RAND USR 16516.

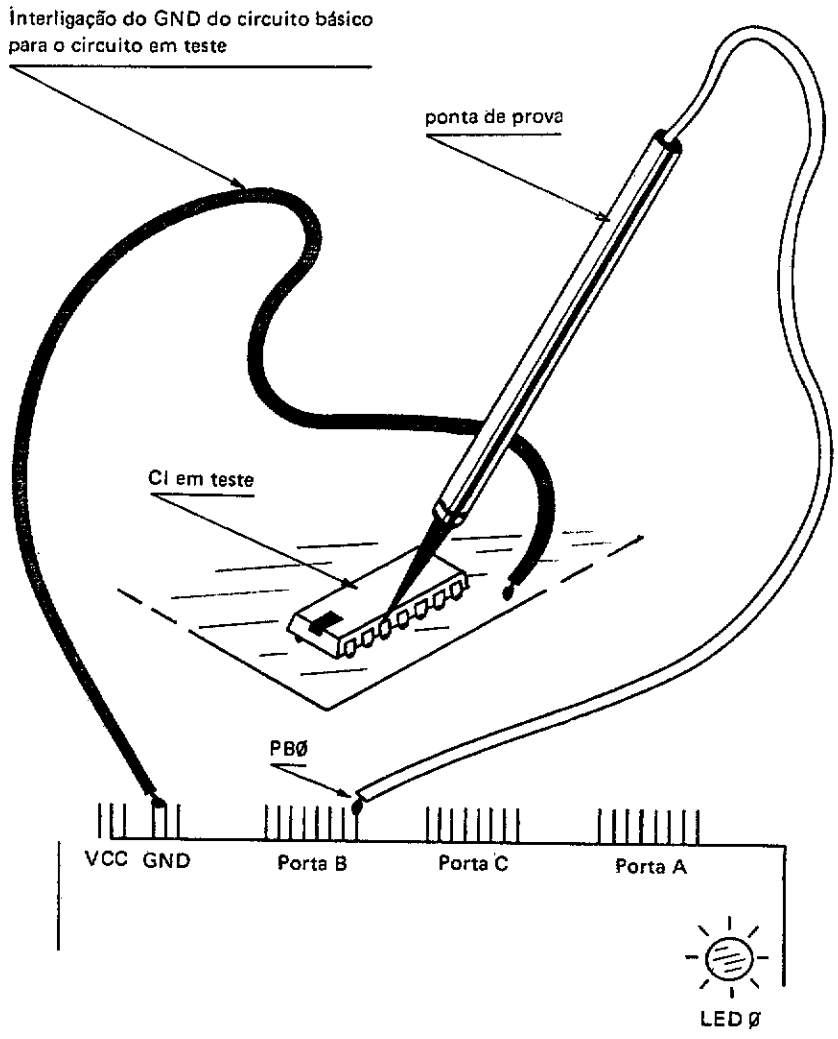


Fig. 13

PONTA LÓGICA

Atenção: para evitar cargas do circuito básico em cima do aparelho a ser testado, convém desligar todos os resistores de 1 kΩ das linhas do PIO que serão usadas.

Este projeto permite medir os níveis lógicos TTL e ter um controle visual do estado deste nível (0 ou 1), indicado pelos LED's existentes no circuito básico.

Digite este pequeno programa:

```
ROTINA BASIC 12

10 POKE 8195,130
20 LET A=PEEK 8193
30 POKE 8192,A
40 GOTO 20
```

Se estiver medindo os sinais lógicos dentro do seu micro, ou até na própria placa do 8255A, então basta simplesmente encostar um fio no ponto a ser testado e observar o LED 0. (Fig. 13).

Se o LED 0 acender, normalmente, estará medindo um nível "1", e se permanecer apagado, o nível medido estará em "0". Pode acontecer, que o LED acenda, porém com uma luminosidade bastante reduzida. Neste caso, com certeza estará medindo uma seqüência de sinais mistos, mudando constantemente o nível.

Também é possível testar os níveis lógicos em outros aparelhos. Neste caso é necessário interligar o GND (negativo) do circuito básico com o GND do outro aparelho, para criar desta maneira uma referência comum.

O mesmo efeito, com uma resposta muito mais rápida, pode ser obtido mediante esta pequena rotina:

```
ROTINA ASSEMBL   10     N=13

*16514  3E  02  32  03  20  3F  01  336
 16521  20  32  00  20  18  F6      388
```

Às vezes há necessidade de testar mais níveis lógicos ao mesmo tempo. Neste caso, podem ser usadas as entradas PB0 até PB7 e mantido tanto o programa em BASIC como o de ASSEMBLY acima.

Se quisermos representar os sinais na tela da TV, teremos que usar pelo menos alguma rotina em linguagem de máquina, pois um respectivo programa em BASIC demoraria muito para processar todas as entradas e falsificaria portanto as leituras.

```
ROTINA ASSEMBL  11    N=92

*16514  00  28  04  3E  1D  D7  C9  551
 16521  3E  1C  18  FA  3E  96  32  631
*16528  03  20  3A  00  20  32  02  305
 16535  40  01  00  08  CD  F5  08  539
*16542  3A  82  40  CB  47  CD  83  862
 16549  40  3A  82  40  CB  4F  CD  803
*16556  83  40  3A  82  40  CB  57  737
 16563  CD  83  40  3A  82  40  CB  855
*16570  5F  CD  83  40  3A  82  40  747
 16577  CB  67  CD  83  40  3A  82  894
*16584  40  CB  6F  CD  83  40  3A  836
 16591  82  40  CB  77  CD  83  40  916
*16598  3A  82  40  CB  7F  C3  83  906
 16605  40                      64
```

```
ROTINA BASIC 13

   0 REM    (ROTINA ASSEMBLY 11)
  10 PRINT AT 2,0;"ENTRADAS ""PA
0"" ATE ""PA7"";"
  20 PRINT AT 7,7;"▀▀▀▀▀▀▀▀▀▀▀▀"
  30 RAND USR 16525
  40 PRINT AT 9,7;"▄▄▄▄▄▄▄▄▄▄▄▄"
  50 GOTO 30
```

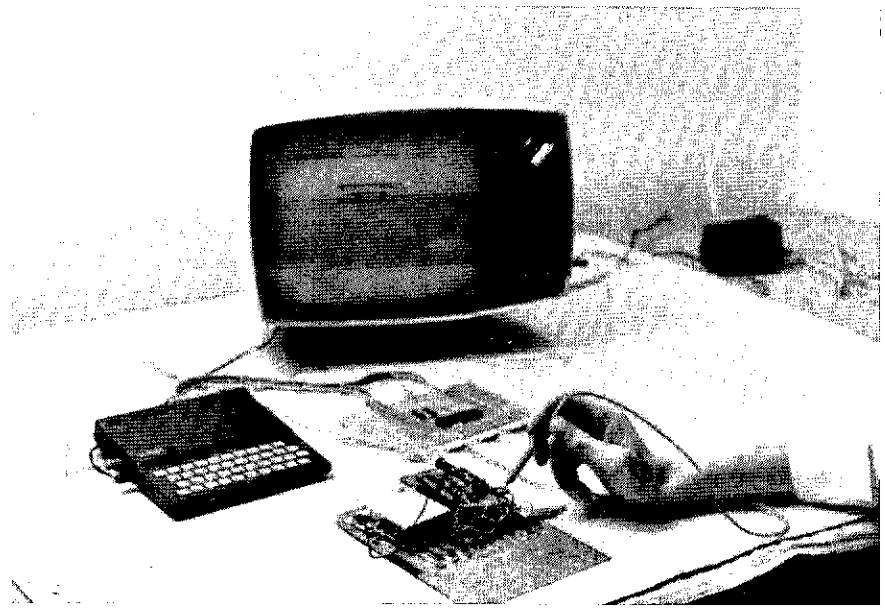



Foto 3

Desta maneira, temos até 8 entradas disponíveis. Em casos particulares é desejável ter até 16 entradas, ou mais, para testar por exemplo, um determinado componente, onde todas as suas entradas e saídas são testadas simultaneamente.

O programa a seguir permite usar todas as 24 linhas como entrada, criando assim uma ponta lógica múltipla.

```
ROTINA ASSEMBLY 12    N=128

*16514  00  00  00  28  04  3E  1D  135
 16521  D7  C9  3E  1C  18  FA  3E  642
*16528  98  32  03  20  3A  00  20  330
 16535  32  82  40  3A  01  20  32  385
*16542  83  40  3A  02  20  32  84  489
 16549  40  01  0F  0A  CD  F5  08  548
*16556  3A  82  40  CD  CB  40  01  725
 16563  0F  0C  CD  F5  08  3A  83  674
*16570  40  CD  CB  40  01  0F  0E  566
 16577  CD  F5  08  3A  84  40  CD  917
*16584  CB  40  C9  F5  CB  47  CD  1192
 16591  85  40  F1  F5  CB  4F  CD  1170
*16598  85  40  F1  F5  CB  57  CD  1178
 16605  85  40  F1  F5  CB  5F  CD  1166
```

```
#16612  35 40 F1 F5 CB 67 CD  ..84
 16619  35 40 F1 F5 CB 6F CD  1208
#16626  35 40 F1 F5 CB 77 CD  ..18
 16633  35 40 F1 CB 7F CD 35  ..09
#16640  40 C9                 3E..
```

ROTINA BASIC 14

```
   0 REM   (ROTINA ASSEMBLY 12
  10 PRINT AT 5,0;"PONTA LOGICA
DE 24 CANAIS:"
  20 PRINT AT 10,0;"PA0 - PA7 --
>"
  30 PRINT AT 12,0;"PB0 - PB7 --
>"
  40 PRINT AT 14,0;"PC0 - PC7 --
>"
  50 RAND USR 16527
  60 GOTO 50
```


ANALISADOR LÓGICO DE ATÉ 24 CANAIS

Atenção: para evitar cargas do circuito básico em cima do aparelho a ser testado, convém desligar todos os resistores de 1 kΩ das linhas do PIO, que serão usadas.

O que é um analisador lógico? Em princípio, este aparelho é uma ponta lógica, de um ou mais canais, que guarda todos os níveis medidos para uma comparação posterior em função do tempo percorrido.

Em nosso caso, vamos efetuar 28 medições sucessivas e listar todos os resultados na tela. Desta maneira, é possível detectar falhas de chaveamento, maus contatos e componentes defeituosos. As entradas, como de costume, são as 24 linhas das portas A, B e C.

ROTINA ASSEMBLY 13 N=158

```
#16514  00 00 00 00 2A 10 40  ...*..@
 16521  01 00 00 00 00 00 ED  .......
#16528  43 00 00 ED 00 47 CD  C....G.
 16535  00 00 00 00 00 00 3A  ......:
#16542  3A 00 00 3A 00 77 3A  :..:.w:
 16549  00 00 77 23 3A FF 00  ..w#:..
#16556  23 C1 10 DF 00 00 00  #......
 16563  40 00 01 00 ED 43 00  @....C.
#16570  40 00 2A 00 C1 44 00  @.*..D.
 16577  00 00 00 64 40 00 01  ...d@..
#16584  06 00 00 C5 ED 5B 00  .....[.
 16591  7E 23 23 23 66 6F 46  ~###foF
#16598  E6 40 C1 10 EF 2A 64  .@...*d
 16605  40 23 22 64 40 C1 10  @#"d@..
```

```
#16512 E2 C9 CB 47 CD 0F 41 955
 16519 CB 4F CD 0F 41 CB 57 857
#16526 CD 0F 41 CB 5F CD 0F 801
 16533 41 CB 67 CD 0F 41 CB 859
#16540 6F CD 0F 41 CB 77 CD 982
 16547 0F 41 CB 7F CD 0F 41 906
#16554 C9 28 0B F5 3E 1D 77 706
 16561 01 21 00 09 09 F5 C9 730
#16568 3E 1C 18 F4                 356
```

ROTINA BASIC 15

```
  0 REM   'ROTINA ASSEMBLY 13
 10 FAST
 20 POKE 8195,155
 30 CLEAR
 40 DIM A$(84)
 50 PRINT AT 11,0;"DIGITE N/L P
ARA INICIAR"
 60 PAUSE 4E4
 70 CLS
 80 RAND USR 16518
 90 POKE 16418,0
100 FOR F=38 TO 40
110 FOR N=28 TO 35
120 PRINT "P";CHR$ F;CHR$ N
130 NEXT N
140 NEXT F
150 SLOW
160 RAND USR 16561
170 GOTO 170
```

Em muitos circuitos a serem analisados, existe geralmente um "clock", que é um trem de pulsos, que por sua vez é responsável pelo sincronismo de todas as funções digitais. Podemos, portanto, usar este clock para encarregá-lo de tomar as leituras nas horas certas, evitando assim leituras iguais.

ROTINA BASIC 16

DEVE SER MANTIDA A ROTINA ASSEMBLY E BASIC ANTERIORES, E ADICIONADAS AS SEQUINTES LINHAS BASIC:


```
41 PRINT "COM TRIGGER ? (S N "
42 INPUT M$
43 PRINT M$
44 IF M$="N" THEN GOTO 48
45 POKE 16540,40
46 POKE 16541,248
47 GOTO 50
48 POKE 16540,0
49 POKE 16541,0
```

Deste modo, de acordo com a seleção, a cada pulso do clock (Trigger), que não deverá ser maior do que 10 MHz, será realizada uma leitura, e após a última leitura listados os resultados na tela.

Até agora vimos só os números Ø e 1 na tela, indicando assim, diretamente, os níveis lógicos. Diminuindo o número máximo das entradas de 24 para 12, podemos representar os níveis lógicos graficamente, criando assim um visual bastante profissional:

ROTINA ASSEMBLY 14 N=168

```
#16514 00 00 00 00 2A 10 40 122
 16521 01 05 00 09 22 82 40 243
#16528 06 10 C5 3A 01 20 E6 552
 16535 08 47 3A 01 20 E6 08 405
#16542 B8 28 F8 3A 00 20 77 681
 16549 23 3A 01 20 77 23 C1 473
#16556 10 E4 C9 2A 40 40 01 564
 16563 05 00 09 22 82 40 06 250
#16570 10 C5 E5 2A 82 40 7E 815
 16577 23 22 82 40 E1 CD E4 921
#16584 40 CD F9 40 E5 2A 82 983
 16591 40 7E 23 22 82 40 E1 576
#16598 CD E4 40 2A 84 40 D8 770
 16605 22 84 40 C1 10 D6 C9 856
#16612 CB 47 CD 0E 41 CB CD 840
 16619 CD 0E 41 CB 57 CD 0E 793
#16626 41 CB 5F CD 0E 41 CB 844
 16633 CB 67 CD 0E 41 CB 6F 904
#16640 CD 0E 41 CB 77 CD 0E 825
 16647 41 CB 7F CD 0E 41 C9 869
#16654 28 0B F5 3E 00 77 23 721
 16661 01 42 00 09 A7 C9 F5 669
#16668 3E 63 18 F3                 460
```

ROTINA BASIC 17

```
  0 REM    ROTINA ASSEMBLY 14
  5 DIM A$(56)
 10 POKE 8195,155
 20 PRINT "COM TRIGGER ?  S/N "
 30 INPUT M$
 40 PRINT M$
 50 IF M$="N" THEN GOTO 90
 60 POKE 16543,40
 70 POKE 16544,248
 80 GOTO 110
 90 POKE 16543,0
100 POKE 16544,0
110 PRINT AT 11,0;"DIGITE N/L P
ARA INICIAR"
120 PAUSE 4E4
130 RAND USR 16518
140 CLS
150 POKE 16418,0
160 FOR F=28 TO 35
170 PRINT "PA";CHR$ F
180 PRINT
190 NEXT F
200 FOR F=28 TO 31
210 PRINT "PB";CHR$ F
220 PRINT
230 NEXT F
240 RAND USR 16559
250 GOTO 250
```

Para criar as curvas, usamos os caracteres conforme a Fig. 14.

Caractere para representar graficamente o

nível Ø =

nível 1 =

Assim, uma seqüência de níveis lógicos ØØ111ØØ1Ø11ØØØ será representada graficamente:

Fig. 14

LIGANDO UM RELÉ

Já vimos as possibilidades do circuito básico, ligando apenas alguns fios nas entradas e saídas das suas três portas. Chegou a hora de ampliar a capacidade deste circuito, ligando relés, que por sua vez podem chavear e comandar qualquer outro aparelho independente.

A ligação de um relé requer poucos componentes, como se pode ver na Fig. 15.

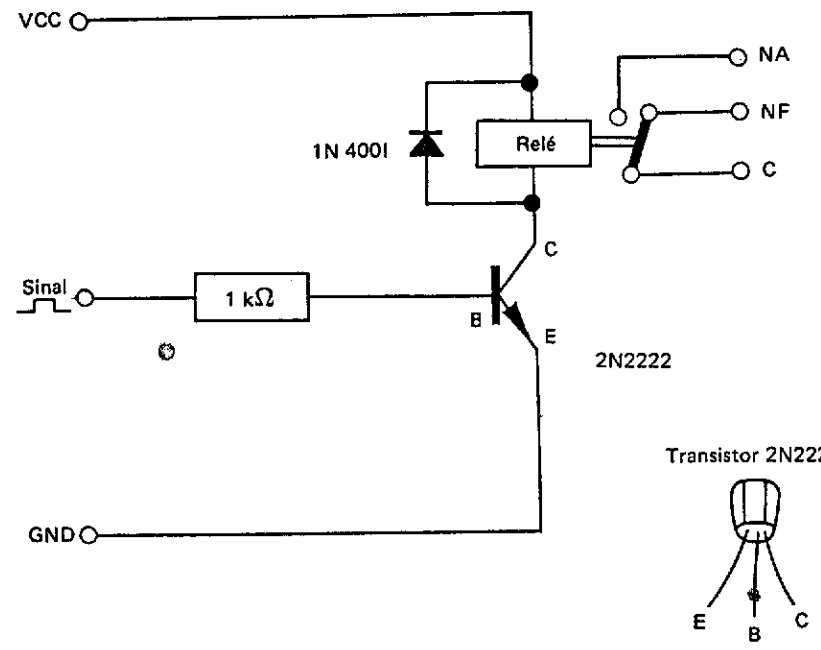


Fig. 15

Lista de material:

1 x transistor 2N2222
1 x resistor 1 kOhm 1/8 Watt
1 x diodo 1N4001
1 x relé para 6 Volts
circuito impresso
cabinho

A montagem não é crítica, e para ampliar a quantidade dos relés deve ser montado o mesmo circuito para cada relé adicional.

Vamos testar o funcionamento do relé ligado em PB0.

ROTINA BASIC 18

```
  10 POKE 8195,128
  20 PRINT AT 11,0;"TESTANDO O R
ELE:"
  30 PRINT AT 14,0;"PARA LIGAR D
IGITE ""L"""
  40 PRINT "PARA DESLIGAR DIGITE
 ""D"""
  50 LET A$=INKEY$
  60 IF A$="L" THEN POKE 8193,1
  70 IF A$="D" THEN POKE 8193,0
  80 GOTO 50
```

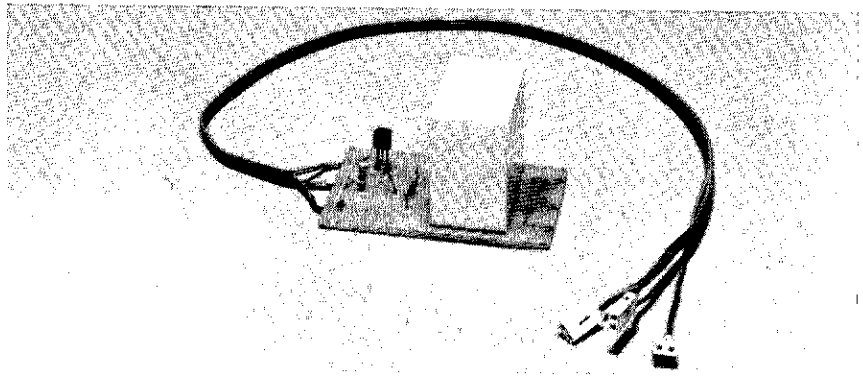


Foto 4

Assim, para ligar um relé ligado em PA0, deve ser feito um "POKE 8192,1" e para acionar um relé na saída PC5, o comando certo será "POKE 8194,32".

DISCADOR DE TELEFONE

O projeto a seguir, já usando o recurso do relé, permitirá discar para qualquer telefone com auxílio do computador. Como o processamento do funcionamento não precisa ser muito rápido, podemos usar sem problemas apenas rotinas escritas em BASIC.

O nosso primeiro programa pede o número do telefone que deve ser digitado. É permitido usar espaços e traços no número, pois o programa BASIC foi escrito de tal maneira que detecta somente os números. Antes de passar para o programa em si, vamos ver como é feita a discagem no telefone. (Fig. 16).

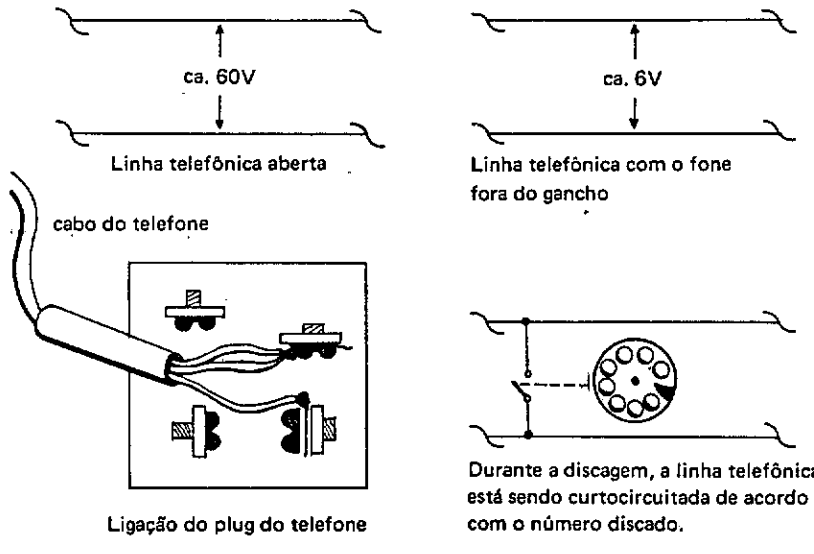


Fig. 16

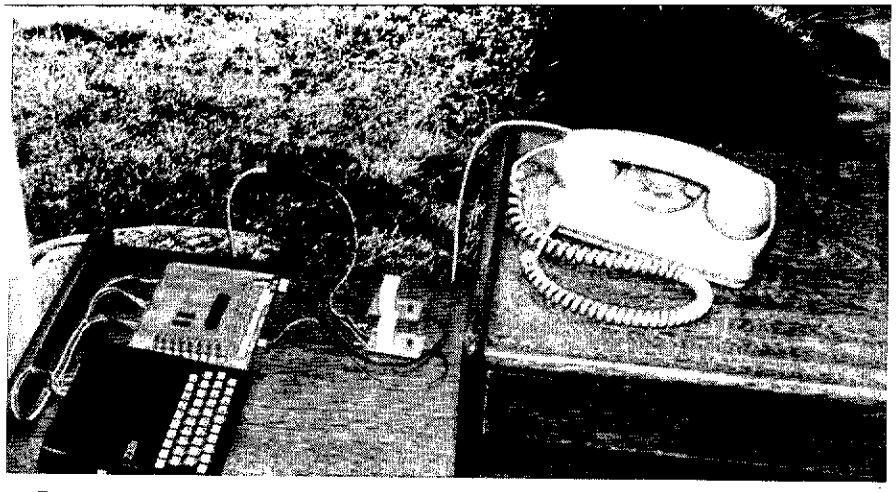


Foto 5

Se o número discado for 5, então será "curto-circuitada" cinco vezes a linha telefônica. Com o número 1 apenas uma vez, e discando 0 há dez curto-circuitos da linha telefônica.

Durante a discagem, o telefone em si deverá ser isolado do circuito, caso contrário, a discagem não se efetuará.

ROTINA BASIC 19

```
  5 SLOW
 10 POKE 8195,128
 20 PRINT "QUAL E O NUMERO? ";
 30 INPUT A$
 40 PRINT A$
 50 LET B$=A$
 60 LET A$=B$
 70 FOR F=1 TO LEN A$
 80 IF A$(F)<"0" OR A$(F)>"9" T
HEN GOTO 180
 90 LET A=CODE A$(F)-28
100 IF A=0 THEN LET A=10
110 FOR G=1 TO A
120 POKE 8192,2
130 PAUSE 5
140 POKE 8192,1
150 NEXT G
160 POKE 8192,0
170 PAUSE 30
180 NEXT F
190 PRINT "QUER REPETIR? (R "
200 INPUT C$
210 IF C$="R" THEN GOTO 60
220 CLS
230 RUN
```


A ligação dos relés e dos seus contatos pode ser vista na Fig. 17.

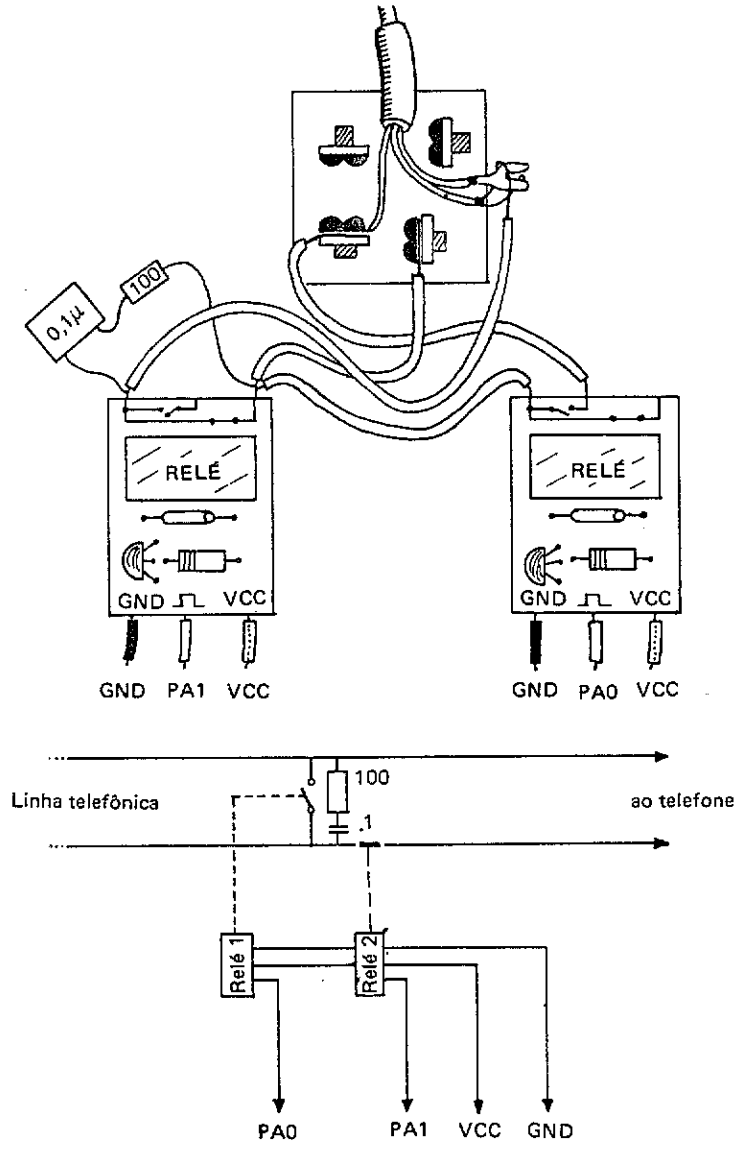


Fig. 17

O programa anterior permite discar até 12 números seguidos, atendendo assim todas as discagens nacionais e internacionais. Se a linha estiver ocupada, pode-se discar novamente apertando a tecla "R". Antes porém de apertar esta tecla, deve-se desligar e ligar novamente o telefone para obter uma linha livre.

Podemos ainda acionar ao discador automático um programa BASIC capaz de armazenar até 300 telefones e nomes diferentes, onde cada nome pode ter até 15 caracteres. Assim, para chamar alguma pessoa ou firma, pode-se digitar o número diretamente, ou ainda entrar com o nome correspondente previamente armazenado.

ROTINA BASIC 20

```
  10 DIM U$(15)
  20 LET V$="
  "
  30 DIM I$(1,32)
  40 DIM A$(300,32)
  50 LET N=0
  60 CLS
  65 POKE 8195,128
  70 PRINT AT 0,7;"AGENDA TELEFO
NICA"
  80 PRINT AT 4,0;"MENU - DIGITE
 O NUMERO:"
  90 PRINT AT 7,0;"1...PROCURAR
PELO NOME"
 100 PRINT AT 9,0;"2...PROCURAR
PELO NUMERO"
 110 PRINT AT 11,0;"3...DISCADOR
"
 120 PRINT AT 13,0;"4...ATUALIZA
R O ARQUIVO"
 130 PRINT AT 15,0;"5...GRAVAR O
 ARQUIVO"
 140 PRINT AT 19,0;N;" NUMEROS C
OMPUTADOS"
 150 LET N$=INKEY$
 160 IF N$="1" THEN GOTO 770
 170 IF N$="2" THEN GOTO 550
 180 IF N$="3" THEN GOTO 240
 190 IF N$="4" THEN GOTO 360
 200 IF N$="5" THEN GOTO 220
 210 GOTO 150
 220 SAVE "ARQUIVO"
 230 GOTO 150
 240 CLS
 250 PRINT AT 10,0;"DIGITE O NUM
ERO QUE DEVO DISCAR:"
 260 INPUT U$
 270 PRINT AT 13,0;"-->";U$
 280 PRINT AT 17,0;"PARA DISCAR
DIGITE ""D"""
 290 INPUT J$
 300 IF J$="D" THEN GOTO 320
 310 GOTO 60
 320 LET J$=U$+V$
 330 LET J$=U$(1 TO 15)
 340 GOSUB 910
 350 GOTO 60
 360 CLS
 370 PRINT "ATUALIZANDO O ARQUIV
O"
 380 PRINT AT 7,0;"ENTRE COM O N
OME DA PESSOA"
 390 INPUT P$
 400 PRINT ,,"-->";P$
 410 PRINT AT 21,0;"PARA RETIFIC
AR DIGITE ""R"""
 420 INPUT W$
 430 IF W$="R" THEN GOTO 360
 440 LET P$=P$+V$
 450 LET I$(1,1 TO 15)=P$(1 TO 1
5)
 460 PRINT AT 11,0;"ENTRE COM O
NUMERO DO TELEFONE"
 470 INPUT P$
 480 PRINT ,,"-->";P$
 490 INPUT W$
 500 IF W$="R" THEN GOTO 360
 510 LET P$=P$+V$
 520 LET I$(1,16 TO 32)=P$(1 TO
32)
 530 LET N=N+1
 540 LET A$(N,1 TO 32)=I$(1,1 TO
32)
 550 GOTO 60
 560 CLS
 570 PRINT AT 7,0;"ENTRE COM O N
UMERO"
 580 INPUT P$
 590 PRINT ,,"-->";P$
 600 PRINT AT 21,0;"PARA RETIFIC
AR DIGITE ""R"""
 610 INPUT W$
 620 IF W$="R" THEN GOTO 560
 630 FAST
 640 FOR F=1 TO N
 650 IF VAL A$(F,16 TO 32)=VAL P
$ THEN GOTO 720
 660 NEXT F
```

```
 670 PRINT AT 15,0 "NUMERO NAO E
NCONTRADO"
 680 PAUSE 300
 690 CLS
 700 SLOW
 710 GOTO 60
 720 PRINT AT 13,0;A$(F,1 TO 32
 730 SLOW
 740 LET U$(1 TO 15)=A$(F,16 TO
32)
 750 PRINT AT 21,0;U$
 760 GOTO 280
 770 CLS
 780 PRINT AT 11,0; ENTRE COM O
NOME"
 790 INPUT P$
 800 PRINT ;"-->";P$
 810 PRINT AT 21,0;"PARA RETIFIC
AR DIGITE ""R"""
 820 INPUT W$
 830 IF W$="R" THEN GOTO 770
 840 LET P$=P$+U$
 850 LET P$=P$(1 TO 15)
 860 FAST
 870 FOR F=1 TO N
 880 IF A$(F,1 TO 15)=P$(1 TO 15
) THEN GOTO 720
 890 NEXT F
 900 GOTO 670
 910 CLS
 920 PRINT AT 11,0;"ESTOU DISCAN
DO O NUMERO"
 930 FOR F=1 TO 15
 940 IF U$(F)<"0" OR U$(F)>"9" T
HEN GOTO 1040
 950 LET A=CODE U$(F)-28
 960 IF A=0 THEN LET A=10
 970 FOR G=1 TO A
 980 POKE 8192,2
 990 PAUSE 5
1000 POKE 8192,1
1010 NEXT G
1020 POKE 8192,0
1030 PAUSE 30
1040 NEXT F
1050 RETURN
```

Para rodar o programa acima pela primeira vez, digite RUN. Para gravar na fita use o SAVE conforme o menu do programa. Se por acaso o programa parar, digite sempre GOTO 60 para reiniciar, caso contrário, poderá perder todos os dados armazenados.

REFLEXÔMETRO

Vimos anteriormente, no capítulo "Tiro ao Alvo", que o programa podia testar a reação do jogador, mas não dava informação do tempo real do reflexo. Este é o objetivo deste projeto, onde será medida a duração do seu reflexo a partir de um estímulo luminoso ou por um sinal acústico.

ROTINA ASSEMBLY 15 N=61

```
#16514 18 0E 00 00 2A 84 40 275
 16521 23 06 20 10 FE 22 84 509
#16528 40 C9 21 00 00 22 84 464
 16535 40 3E 89 32 03 20 3A 406
#16542 02 20 FE 00 20 F9 3E 631
 16549 FF 32 00 20 3A 02 20 429
#16556 FE 00 20 05 CD 86 40 694
 16563 18 F4 ED 4B 84 40 3E 836
#16570 00 32 00 20 C9          283
```

ROTINA BASIC 21

```
  0 REM   (ROTINA ASSEMBLY 15)
 10 RAND
 20 LET S=0
 30 FOR F=1 TO 5
 40 LET X=INT (RND*150)+60
 50 FOR G=1 TO X
 60 NEXT G
 70 LET A=USR 16514
 80 PRINT "TESTE NR. ";F;": ";A
;" MS"
 90 LET S=S+A
100 NEXT F
110 PRINT "SUA MEDIA FOI :",,S/
5;" MILISEGUNDOS."
```

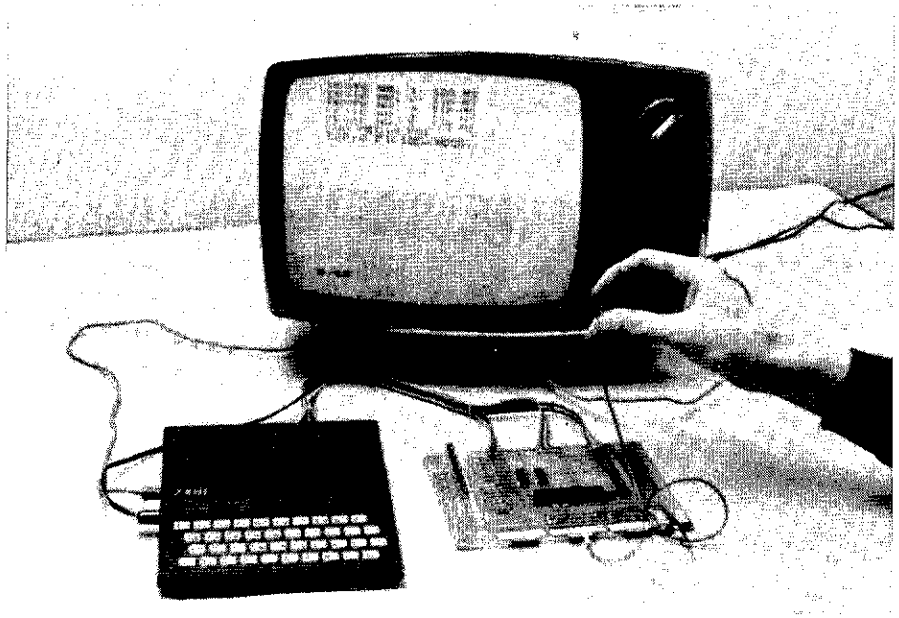


Foto 6

O funcionamento é simples: no programa acima, após um tempo de espera de duração aleatória, acender-se-ão todos os LED's no circuito básico. O mais rapidamente que puder, o jogador deve apertar o botão conectado na linha PCØ. São feitos cinco testes, após os quais será calculada a duração média do seu reflexo, com indicação dos tempos individuais. A rotina ASSEMBLY já está calibrada para contar o tempo em milissegundos (o programa deve trabalhar em SLOW).

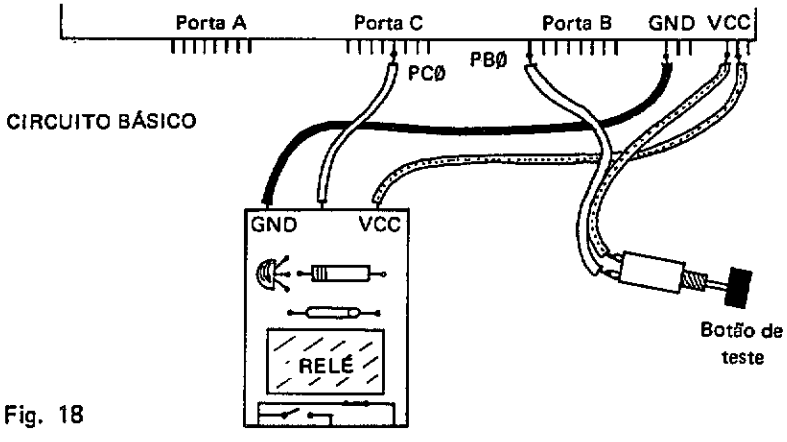


Fig. 18

O segundo programa, de funcionamento semelhante, usa agora um relé ao invés dos LED's. Este relé será conectado na linha PCØ e o botão na linha PBØ (Fig. 18).

A rotina BASIC para a rotina ASSEMBLY a seguir é a mesma como no caso dos LED's acima.

```
ROTINA ASSEMBL   1E     N=42


*16514 3E 02 32 03 20 21 00 310
 16521 00 3A 01 20 FE 00 20 377
*16528 F9 3E 01 32 02 20 3A 454
 16535 01 20 FE 00 20 07 23 361
*16542 06 20 10 FE 1E F2 44 642
 16549 4D 3E 00 32 02 20 C9 424
```

Caso se queira aumentar a quantidade dos testes, obtendo assim uma precisão maior do valor médio do seu reflexo, devem ser alteradas as linhas 30 e 110 respectivamente.

Exemplo:

mudar a rotina para 10 testes:

mudar a linha 30 para

```
30  FOR F = 1 TO 10
```

e a linha 110 para

```
110  PRINT "SUA MÉDIA FOI:",,S/10;" MILISSEGUNDOS"
```

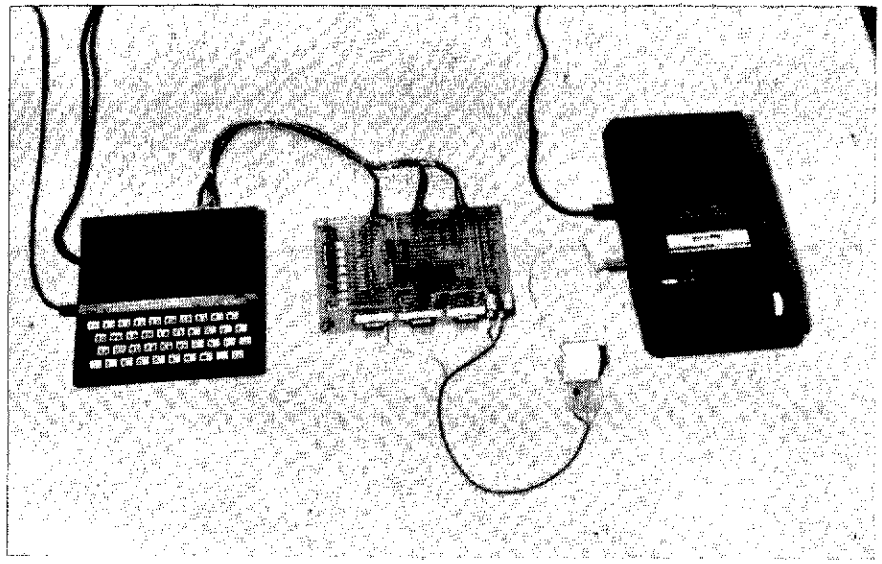


Foto 7

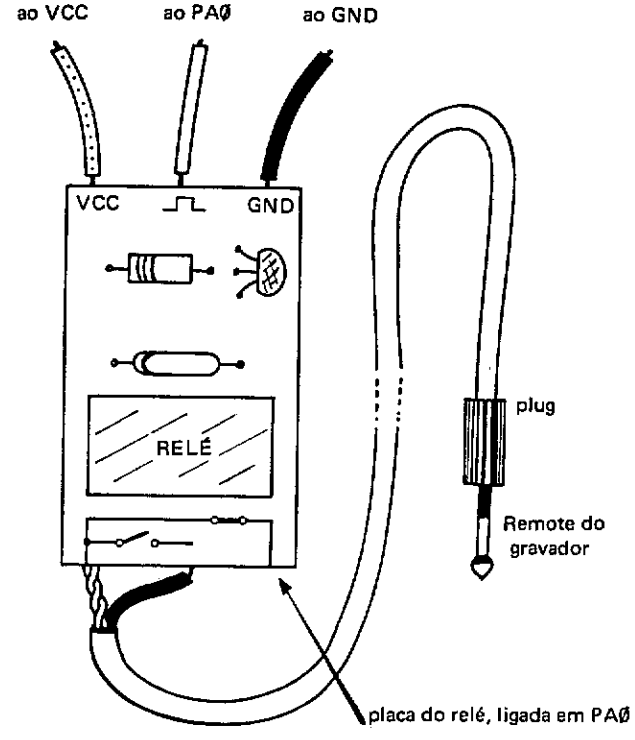


Fig. 19

CONTROLANDO O GRAVADOR

Com um relé conectado ao nosso circuito básico, podemos controlar o nosso gravador durante a leitura e a gravação de programas, dispensando assim a necessidade de ligar e desligar o gravador o tempo todo. Com isto ganharemos mais espaço na fita, pois podemos eliminar ainda aproximadamente 6 segundos de silêncio antes de cada gravação.

O seu gravador deve ter uma entrada "Remoto", que liga e desliga o motor dentro do gravador. Este motor agora será controlado por um relé, ligado na saída PA0 do circuito básico.

Para o SAVE, escreva o seguinte programa em linguagem de máquina, e substitua o comando SAVE por RAND USR 16514.

```
ROTINA ASSEMBLY 17   N=38

*16514 CD 23 0F CD 23 0F 3E 572
 16521 80 32 03 20 3E 01 32 326
*16528 00 20 06 00 C5 06 00 241
 16535 10 FE C1 10 F8 CD 0B 943
*16542 03 3E 00 32 00 20 CD 352
 16549 2B 0F C9                259
```

e, para o LOAD teremos esta rotina:

```
ROTINA ASSEMBLY 18   N=25

*16514 CD 23 0F 3E 80 32 03 498
 16521 20 3E 01 32 00 20 CD 382
*16528 44 03 3E 00 32 00 20 215
 16535 CD 2B 0F C9             464
```

que será iniciada também com RAND USR 16514.

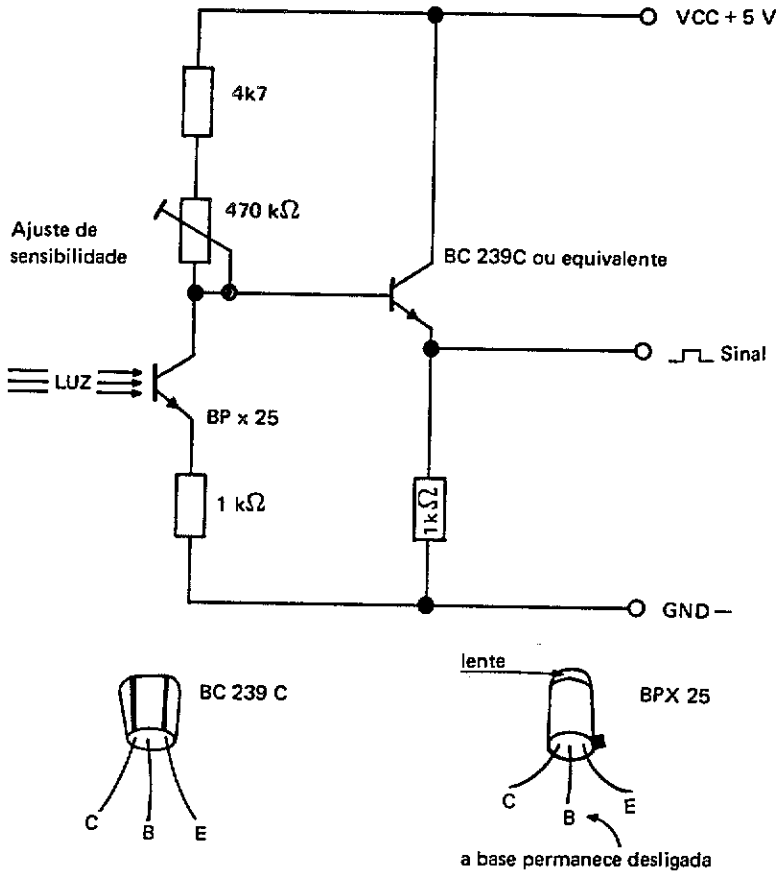


Fig. 20

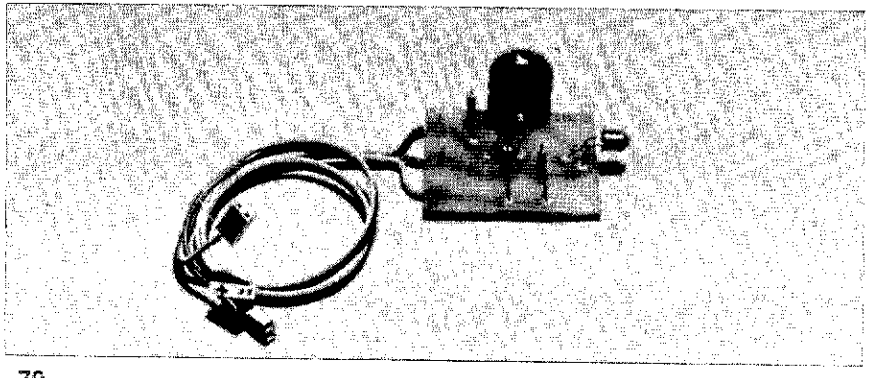


Foto 8

BARREIRA DE LUZ

Não estamos limitados em ligar apenas relés, LED's e botões ao nosso circuito básico, pois existem outros periféricos bastante interessantes, abrindo novas perspectivas.

Como exemplo, podemos montar um sensor sensível à luz, usando o circuito conforme a Fig. 20.

MATERIAL NECESSÁRIO PARA MONTAGEM:

1 x trimpot 470 kohms
2 x resistor 1 kohms 1/4 watt
1 x resistor 4,7 kohms 1/4 watt
1 x transistor BC 239C ou equivalente
1 x fototransistor BPX 25
circuito impresso
cabinho

No escuro, o fototransistor apresenta uma resistência interna elevada, correspondendo a um nível lógico 1 na saída do circuito da Fig. 20. A partir de uma certa luminosidade, a resistência interna do fototransistor cai para alguns ohms, mudando o nível lógico do circuito de 1 para 0.

Ligando este sensor à linha PC0, podemos testar este circuito com o seguinte programa:

ROTINA BASIC 22

```
10 POKE 8195,137
20 IF PEEK 8194=0 THEN GOTO 50
30 PRINT AT 11,12;"ESCURO"
40 GOTO 20
50 PRINT AT 11,12;"HA LUZ"
60 GOTO 20
```

Agora, escurecendo o fototransistor com o dedo, aparecerá na tela a mensagem "ESCURO", e destapando o fototransistor, a mensagem muda para "HÁ LUZ". A sensibilidade do circuito pode ser regulada em ampla escala por intermédio do trimpot de 470 kohms.

Montando o fototransistor dentro de um tubo de papelão, para evitar influência da luz do ambiente, podemos juntamente com uma fonte de luz dirigida para o fototransistor, montar uma eficiente barreira de luz (Fig. 21), controlando por exemplo a entrada principal da nossa casa.

O sensor está ligado em PC0, e na linha PA0 pode ser ligado um relé para acionar por exemplo um alarme. O programa correspondente é simples:

ROTINA BASIC 23

```
10 POKE 8195,137
20 IF PEEK 8194=0 THEN GOTO 20
30 POKE 8192,1
40 STOP
```

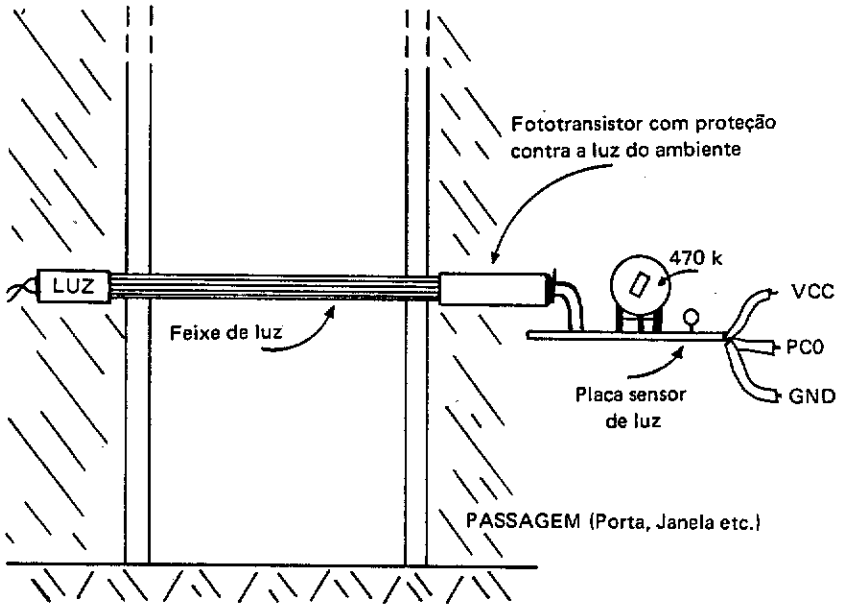


Fig. 21

CONTROLADOR DE EVENTOS

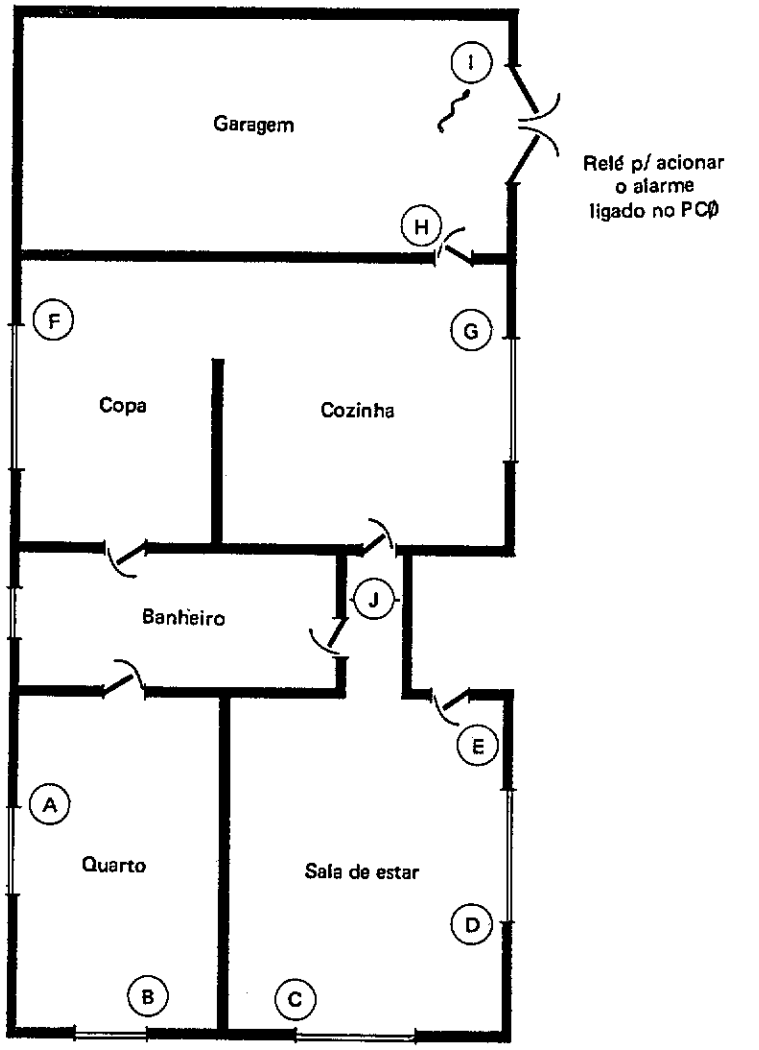
O objetivo deste projeto é de contar eventos, como por exemplo a quantidade de peças numa esteira, pessoas passando por uma porta, etc.

No exemplo a seguir, qualquer uma das entradas da porta C serve como entrada, enquanto a linha PA0 está reservada para chavear um relé assim que a contagem chega a um valor pré-definido.

ROTINA BASIC 24

```
  10 POKE 8195,137
  20 POKE 8192,0
  30 PRINT "ENTRE COM O NUMERO L
IMITE"
  40 INPUT A
  50 LET N=0
  60 PRINT "DIGITE N/L PARA INIC
IAR"
  70 IF INKEY$=""  THEN GOTO 70
  80 FAST
  90 IF PEEK 8194=0 THEN GOTO 90
 100 IF PEEK 8194<>0 THEN GOTO 1
00
 110 LET N=N+1
 120 IF N=A THEN GOTO 140
 130 GOTO 90
 140 CLS
 150 POKE 8192,1
 160 PRINT AT 11,0;"CONTAGEM MAX
IMA ATINGIDA"
 170 SLOW
 180 STOP
```

A velocidade de resposta com o programa acima é de aproximadamente 4 eventos por segundo e quando há necessidade de velocidades maiores, nada impede que usemos uma rotina equivalente, escrita em linguagem de máquina.



Botões, chaves ou reed-switches — Sensor de luz — Barreira de luz

A B C D E F G H — I — J

PA0 PA1 PA2 PA3 PA4 PA5 PA6 PA7 — PBØ — PB1

Fig. 22

ALARME RESIDENCIAL

Combinando os recursos dos sensores de luz, chaves e relés, podemos montar um eficiente alarme residencial, que além de proteger a sua casa dos ladrões, ainda pode supervisionar o consumo da luz, e com sensores apropriados, também o consumo de água e gás. Paralelamente ainda poderá incluir um aviso de incêndio.

As linhas PA0 até PA7 e PB0 até PB7 representam 16 entradas dos diversos sensores, enquanto que as linhas PC0 até PC7 podem chavear diversos relés.

A Fig. 22 indica uma sugestão de alarme residencial, onde podemos ver as barreiras de luz, os sensores nas portas e janelas, os sensores de luz para controlar a luz acesa nas dependências, bem como o alarme em si.

É evidente, que devemos proteger o funcionamento perfeito do microcomputador e do circuito básico, pois mesmo com uma falta de luz, o funcionamento deverá ser garantido. Para manter o alarme mesmo nestas condições, faz-se necessário o emprego de uma bateria de 12 Volts, que continua alimentando o micro, os sensores e o alarme durante a falta de luz. Retornando a luz da rede, a bateria será desligada automaticamente.

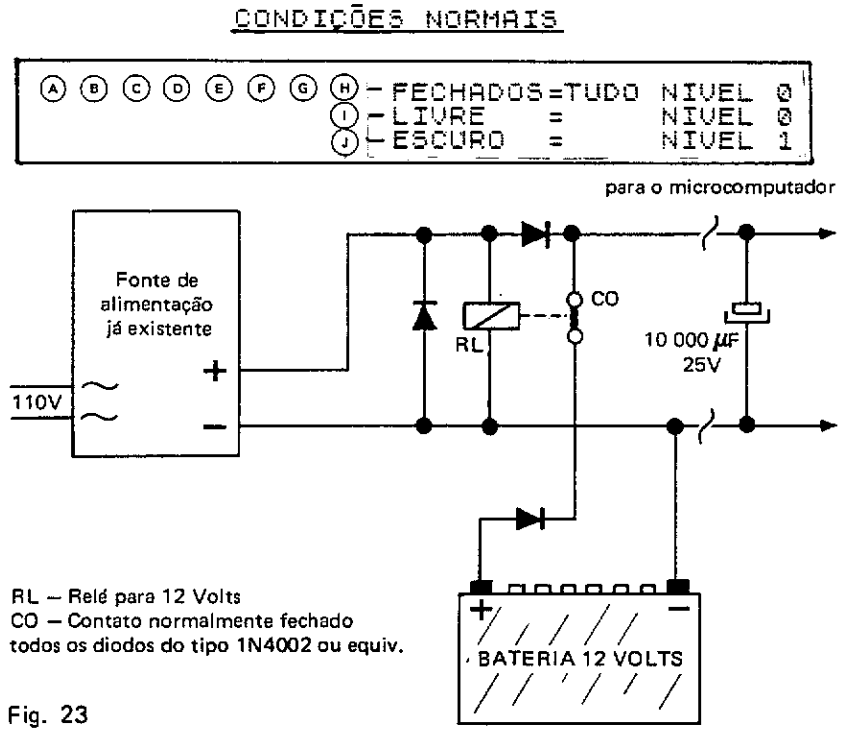
O programa para o exemplo de alarme residencial acima é o seguinte:

ROTINA BASIC 25

```
10 POKE 8195,146
20 LET A=0
30 PRINT "PARA ACIONAR O ALARM
E DIGITE ""A"""
40 IF INKEY$="A" THEN LET A=1
50 PRINT AT 11,0;"ALARME ";
60 IF A=0 THEN PRINT "DESLIGAD
```

```
0"
  70 IF A=1 THEN PRINT 'ACIONADO
"
  80 LET Y=0
  90 IF PEEK 8192<>0 THEN LET Y=
1
 100 IF PEEK 8193<>2 THEN LET Y=
1
 110 IF Y=1 AND A=1 THEN GOTO 13
0
 120 GOTO 40
 130 POKE 8194,1
 140 STOP
```

Podemos ainda adicionar o discador automático de telefone, junto com um gravador e uma fita preparada, que liga automaticamente para um parente ou a polícia, assim que alguém tentar entrar na casa. É evidente, que neste caso deverá existir um tempo mínimo de 20 segundos de espera, para que o circuito possa ser desativado a tempo pelo proprietário da casa.



RL – Relé para 12 Volts
CO – Contato normalmente fechado
todos os diodos do tipo 1N4002 ou equiv.

Fig. 23

TEM ALGUÉM EM CASA?

Apesar de não se tratar de um alarme residencial, este projeto também proteje a sua casa de eventuais ladrões, utilizando um efeito psicológico.

Quando estiver viajando, o perigo da sua casa ser assaltada é muito grande, pois poucas pessoas têm um caseiro ou uma outra pessoa, que possa cuidar da casa. Assim, durante a noite toda a sua casa permanece com as luzes apagadas, e durante o dia não se escuta nenhum ruído. A conclusão, que não tem ninguém em casa, é óbvia.

O segredo é simular dentro de uma casa vazia, as atividades rotineiras, evitando simulações sem valia, como com uma lâmpada acesa e o rádio ligado, dia e noite.

Com ajuda do nosso circuito básico podemos ligar e desligar as luzes, o rádio e a TV, em intervalos irregulares, deixando os ladrões na certeza de que há pessoas em casa.

Desta vez, usaremos apenas a porta A como entrada, e as portas B e C para chavear os relés. A entrada é necessária para controlar a luz externa da casa, reconhecendo se é dia ou noite. Pois não há lógica de ligar as luzes ao meio dia, ou ligar a TV às cinco horas da manhã.

Como sugestão, podemos distribuir o sensor e os relés da seguinte maneira:

PA0 – sensor de luz, instalado numa janela longe de iluminação de rua
PB0 – relé luz da sala
PB1 – relé luz do quarto
PB2 – relé luz da cozinha
PB3 – relé luz do banheiro
PB4 – relé luz fora da casa
PB5 – relé televisor
PB6 – relé rádio
PB7 – relé alarme (opcional)

PCØ até PC7 — livres para outras finalidades, por exemplo os sensores de alarme residencial

Também aqui se faz necessário o emprego de uma bateria para manter o computador funcionando mesmo com falta de luz. O alarme não deverá disparar indefinidamente, pois não haverá ninguém para desligá-lo. Basta tocá-lo por 30 segundos, que é suficiente para espantar eventuais ladrões.

ROTINA BASIC 25

```
  10 GOSUB 100
  20 POKE 8195,144
  30 GOSUB 6000
  40 GOTO 1000
 100 LET LS=0
 110 LET LB=0
 120 LET LC=0
 130 LET L0=0
 140 LET LF=0
 150 LET TV=0
 160 LET RA=0
 170 LET TEE=0
 180 LET TEC=0
 190 RETURN
1000 LET SEN=0
1010 GOSUB 100
1020 GOSUB 3000
1030 LET A=PEEK 8192
1040 IF A<>0 THEN GOTO 2000
1050 GOSUB 4000
1060 GOSUB 5000
1070 LET TEC=TEC+1
1080 IF RA<>0 THEN LET RA=RA-1
1090 IF TV<>0 THEN LET TV=TV-1
1100 IF RA=0 AND TV=0 AND TEC>=3
0 THEN GOSUB 1130
1110 IF RA=0 AND TV=0 AND TEC>=2
40 THEN GOSUB 1170
1120 GOTO 1020
1130 LET U=RND*101
1140 IF U>15 THEN RETURN
1150 LET RA=INT (RND*30+60
1160 RETURN
1170 LET U=RND*101
1180 IF U>5 THEN RETURN
1190 LET TV=INT (RND*75+15
1200 RETURN
2000 LET SEN=1
2010 GOSUB 100
2020 GOSUB 3000
2030 LET A=PEEK 8192
2040 IF A=0 THEN GOTO 1000
2050 GOSUB 4000
2060 GOSUB 5000
2070 LET TEE=TEE+1
2080 IF LS<>0 THEN LET LS=LS-1
2090 IF LC<>0 THEN LET LC=LC-1
2100 IF LB<>0 THEN LET LB=LB-1
2110 IF L0<>0 THEN LET L0=L0-1
2120 IF LF<>0 THEN LET LF=LF-1
2130 IF TV<>0 THEN LET TV=TV-1
2140 IF RA<>0 THEN LET RA=RA-1
2150 IF LS=0 AND TEE<=150 THEN G
OSUB 2230
2160 IF L0=0 AND TEE<=150 THEN G
OSUB 2270
2170 IF LC=0 AND TEE<=150 THEN G
OSUB 2310
2180 IF LB=0 THEN GOSUB 2350
2190 IF LF=0 AND TEE<=150 THEN G
OSUB 2390
2200 IF TV=0 AND RA=0 AND TEE<=1
50 THEN GOSUB 2430
2210 IF RA=0 AND TV=0 AND TEE<=1
50 THEN GOSUB 2470
2220 GOTO 2020
2230 LET U=RND*101
2240 IF U<20 THEN RETURN
2250 LET LS=INT (RND*50+10)
2260 RETURN
2270 LET U=RND*101
2280 IF U>7 THEN RETURN
2290 LET L0=INT (RND*6+2)
2300 RETURN
2310 LET U=RND*101
2320 IF U>7 THEN RETURN
2330 LET LC=INT (RND*10+5)
2340 RETURN
2350 LET U=RND*101
2360 IF U>15 THEN RETURN
2370 LET LB=INT (RND*4+1)
2380 RETURN
2390 LET U=RND*101
2400 IF U>20 THEN RETURN
2410 LET LF=INT (RND*50+10)
2420 RETURN
```

```
2430 LET U=RND*101
2440 IF U>20 THEN RETURN
2450 LET TV=INT (RND*50+30)
2460 RETURN
2470 LET U=RND*101
2480 IF U>5 THEN RETURN
2490 LET RA=INT (RND*45+15)
2500 RETURN
3000 FOR K=1 TO 4270
3010 NEXT K
3020 RETURN
4000 LET OUT=0
4010 IF LS<>0 THEN LET OUT=OUT+1
4020 IF L0<>0 THEN LET OUT=OUT+2
4030 IF LC<>0 THEN LET OUT=OUT+4
4040 IF LB<>0 THEN LET OUT=OUT+8
4050 IF LF<>0 THEN LET OUT=OUT+1
6
4060 IF TV<>0 THEN LET OUT=OUT+3
2
4070 IF RA<>0 THEN LET OUT=OUT+6
4
4080 POKE 8193,OUT
4090 RETURN
5000 PRINT AT 7,0;
5010 LET X=LS
5020 GOSUB 5180
5030 LET X=L0
5040 GOSUB 5180
5050 LET X=LC
5060 GOSUB 5180
5070 LET X=LB
5080 GOSUB 5180
5090 LET X=LF
5100 GOSUB 5180
5110 LET X=TV
5120 GOSUB 5180
5130 LET X=RA
5140 GOSUB 5180
5150 PRINT AT 15,5;"CLARO "
5160 IF SEN<>0 THEN PRINT AT 15,
5;"ESCURO"
5170 RETURN
5180 LET U$="X......."
5190 IF X=0 THEN LET U$=".......
.X"
5200 PRINT TAB 18;U$
5210 RETURN
6000 PRINT AT 2,0;"CONTROLANDO A
 CASA"
6010 PRINT "------------------"
6020 PRINT AT 5,0; LOCAL ;TAB 16
;"LIGADO DESLIGADO
6030 PRINT AT 7,0  LUZ SALA
6040 PRINT "LUZ QUARTO"
6050 PRINT "LUZ COZINHA"
6060 PRINT "LUZ BANHEIRO
6070 PRINT "LUZ FORA"
6080 PRINT "TV"
6090 PRINT "RADIO
6100 PRINT AT 15,0;"ESTA"
6110 RETURN
```

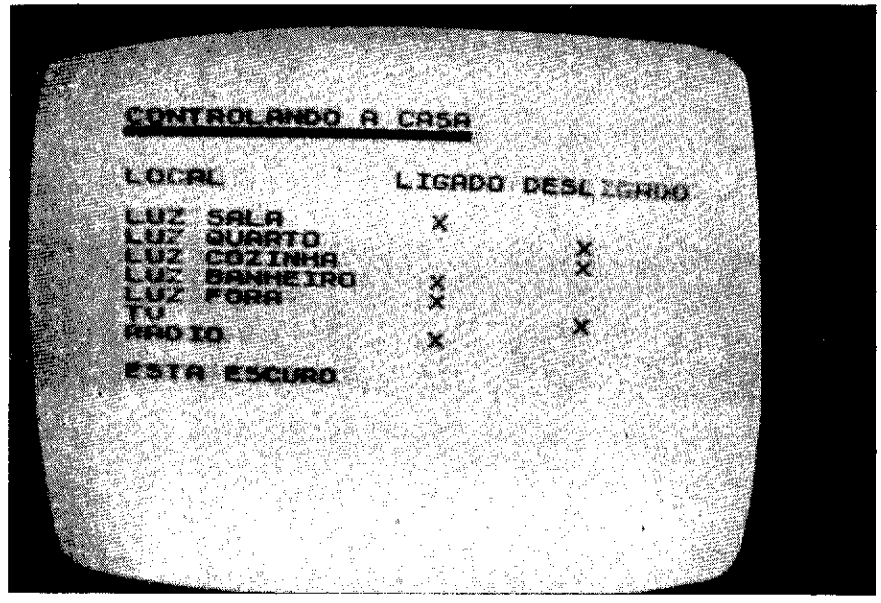


Foto 9

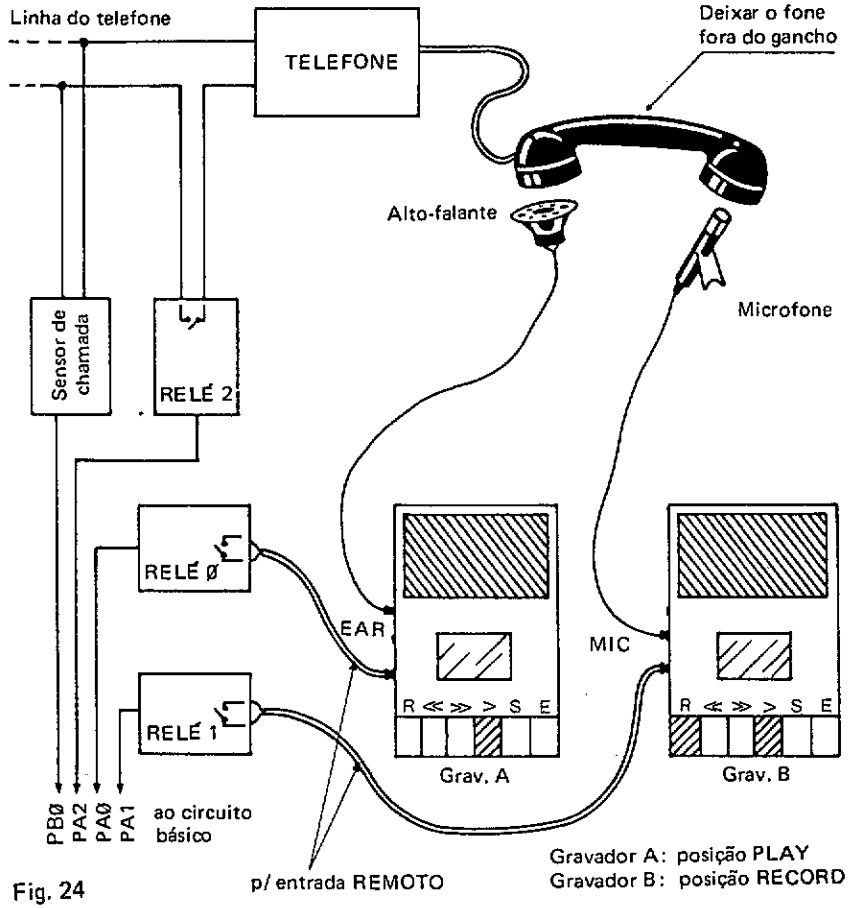
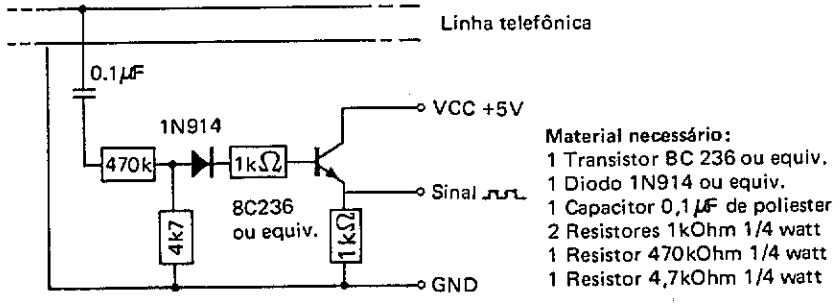


Fig. 24



Material necessário:
1 Transistor BC 236 ou equiv.
1 Diodo 1N914 ou equiv.
1 Capacitor 0,1 µF de poliester
2 Resistores 1kOhm 1/4 watt
1 Resistor 470kOhm 1/4 watt
1 Resistor 4,7kOhm 1/4 watt

Fig. 25

SECRETÁRIA ELETRÔNICA

Uma outra vantagem do nosso circuito básico é o fator econômico, pois podem ser montados aparelhos com resultados iguais ou até superiores a outros equipamentos criados para esta única função, custando muitas vezes mais do que nós empregamos. Um exemplo típico é uma secretária eletrônica, que permite a você atender, gravar e transmitir telefonemas; mesmo não estando em casa.

É um projeto básico, que pode ser ampliado conforme a sua criatividade e necessidade. Trataremos o circuito e os programas necessários para atender e gravar telefonemas, sem a sua interferência. Em conjunto com o discador automático é perfeitamente possível transmitir o recado recebido imediatamente para seu escritório, ou outro lugar onde estiver.

A ligação dos relés na linha telefônica pode ser vista na Fig. 24.

Temos um sensor adicional, que aciona o programa quando toca o telefone, pois em condições normais está presente uma tensão de corrente contínua na linha telefônica, e durante a chamada uma tensão alternada, que será indicada pelo nosso sensor. Este sensor está ligado na porta PB0, e os relés em PA0 e PA1 controlam os dois gravadores, enquanto o relé em PA2 controla a linha telefônica.

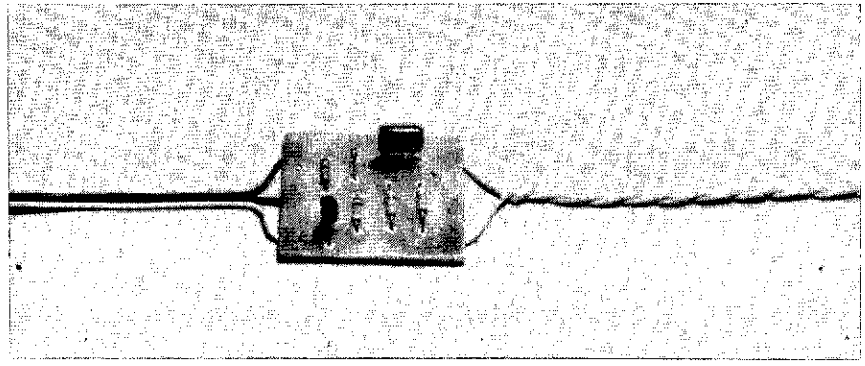


Foto 10

Assim sendo, ao tocar o telefone, o relé PA0 acionará o gravador com a mensagem, e o relé PA2 "atende" o telefone. Após transmitir a mensagem, o relé PA0 desliga, e será ligado o relé PA1, que por sua vez liga o segundo gravador para a gravação do recado. Após alguns segundos, o relé PA1 pára o segundo gravador, e o relé PA2 "desliga" o telefone, aguardando uma nova chamada.

Para efeito de sincronismo da sua mensagem, e o tempo, durante o qual o primeiro gravador (relé PA0) permanece ligado, o programa foi dividido em duas partes.

A primeira parte serve para você gravar as mensagens na fita, enquanto o computador liga e desliga o gravador em tempos certos.

ROTINA BASIC 27

```
 10 POKE 8195,128
 20 POKE 8192,1
 30 PAUSE 400
 40 POKE 8192,0
 50 PAUSE 150
 60 PRINT AT 11,11;"FALE AGORA"
 70 POKE 8192,1
 80 PAUSE 1000
 90 POKE 8192,0
100 CLS
110 GOTO 50
```

Pegue uma fita, de preferência virgem, e coloque-a no gravador totalmente no início. Conecte o plug Remoto (relé ligado em PA0), digite RUN e aguarde a instrução do micro para gravar suas mensagens. O programa automaticamente faz avançar a fita durante a gravação, deixando uma margem de silêncio na mesma. Você deve falar somente quando aparecer a instrução "FALE AGORA" na tela do seu TV. Desta maneira, é possível gravar a fita inteira, garantindo o sincronismo durante o funcionamento da secretária eletrônica.

Se achar que o tempo de gravação de sua mensagem é comprido ou curto demais, poderá alterar o valor da PAUSE na linha 80 do programa anterior, mas deve alterar da mesma maneira a linha 140 da rotina principal a seguir:

ROTINA BASIC 28

```
 10 POKE 8195,130
 20 POKE 8192,3
 30 PAUSE 400
 40 POKE 8192,0
 50 LET A=0
 60 IF PEEK 8193=0 THEN GOTO 60
 70 LET A=A+1
 80 IF A=3 THEN GOTO 110
 90 IF PEEK 8193<>0 THEN GOTO 9
0
100 GOTO 60
110 POKE 8192,4
120 PAUSE 50
130 POKE 8192,5
140 PAUSE 1000
150 POKE 8192,6
160 PAUSE 3000
170 GOTO 40
```

Para testar o sensor da chamada, peça a um amigo que lhe telefone, já com o seguinte programa BASIC rodando no seu micro:

ROTINA BASIC 29

```
 10 POKE 8195,155
 20 LET A=0
 30 PRINT AT 11,8;"NAO ESTA TOC
ANDO"
 40 IF PEEK 8193=0 THEN GOTO 40
 50 LET A=A+1
 60 IF A=3 THEN GOTO 90
 70 IF PEEK 8193<>0 THEN GOTO 7
0
 80 GOTO 30
 90 PRINT AT 11,8;"  "
```

Se, ao tocar o telefone, aparecer a mensagem "ESTA TOCANDO", então o circuito está funcionando perfeitamente. O micro deve estar em SLOW, e a mensagem "ESTA TOCANDO" somente aparecerá após o segundo ou terceiro toque. Desta maneira evitamos um acionamento da secretária eletrônica a partir de uma única interferência na linha telefônica. Resta testar

os relés antes de entrar em funcionamento a rotina principal. Para isto, digite os seguintes comandos diretos:

POKE 8192,130 — inicializa o PIO
POKE 8192,1 — liga o gravador das mensagens (PAØ)
POKE 8192,2 — liga o gravador dos recados (PA1)
POKE 8192,4 — liga o relé conectado a linha telefônica (PA2)

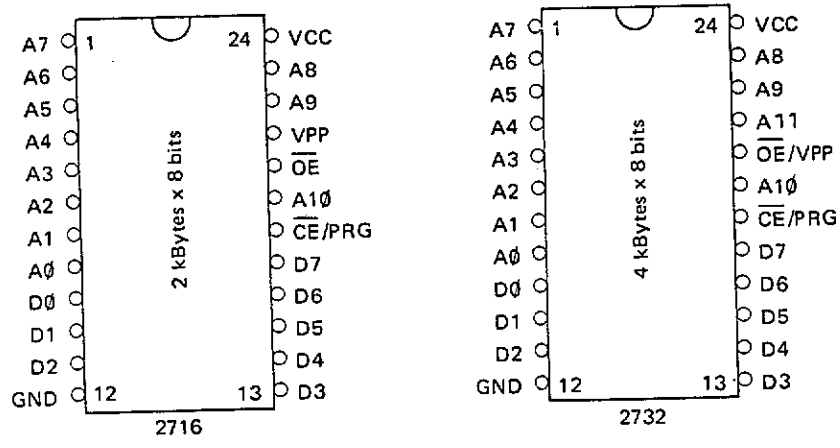
Uma vez testados todos os componentes, coloque uma fita virgem no gravador nº 2, conecte o plug do relé 1 no remoto e aperte as teclas para a gravação. O gravador deverá permanecer parado

A seguir, coloque no gravador em PA0 a sua fita com as mensagens totalmente no início e, após conectar o Remote (relé 0), ligue a tecla para a reprodução. Digitando RUN, os gravadores em PA0 e PA1 deverão ligar imediatamente e desligar após alguns segundos (para posicionar as fitas). Pronto, sua secretária eletrônica está pronta para receber e gravar os telefonemas.

GRAVADOR DE EPROM

Este projeto talvez seja a montagem mais útil para o "hobbista", pois permitirá gravar seus próprios EPROM's do tipo 2716 (de 2 kBytes de memória) e do tipo 2732 (de 4 kBytes de memória).

A diferença de gravação destes dois EPROM's é o nível lógico de programação, que no tipo 2716 é ativo com nível "Ø", e no 2732 ativo em "1". Temos ainda a pinologia com pequenas diferenças, que podem ser vistas na Fig. 26.



AØ - A1Ø — endereçamento até 2Ø47 (com A11 até 4Ø95)
DØ - D7 — linha de dados
OE — output enable
CE/PRG — chip enable/programação

Fig. 26

Para gravar estas EPROM's, são necessárias as seguintes condições:

a) o endereço, onde deve ser gravado o dado, deve estar estabilizado (A0 até A10 no 2716, e A0 até A11 no 2732)

b) o dado deve estar presente nas linhas D0 até D7

c) uma tensão de 25,0 Volts regulada deve estar conectada ao pino de VBB (pino 20 no 2732, e pino 21 no 2716)

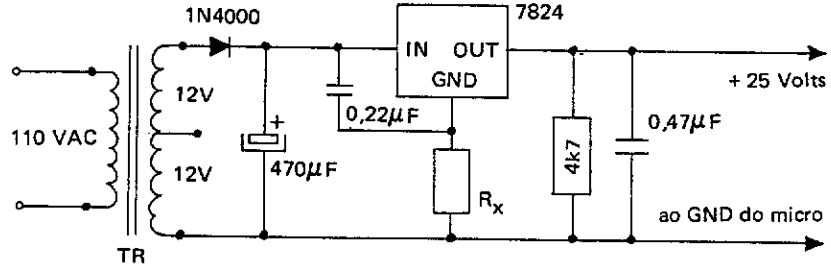
d) o pulso de programação deve ter uma duração de exatamente 50 milissegundos.

Este pulso de programação não pode ser muito curto, pois não garantiria uma programação eficiente, enquanto um pulso comprido demais poderá danificar a EPROM.

Poderíamos controlar o tempo com o 74LS121, mas qualquer falha de componente facilmente alteraria o tempo de 50 milissegundos, colocando assim em perigo a EPROM a ser gravada.

Neste projeto, estamos controlando este tempo por software, que devido ao clock alto do microcomputador apresenta uma precisão total.

No que se refere à tensão de 25 Volts, podemos montar o seguinte circuito (Fig. 27):



Relação do material:
1 Regulador de tensão 7824
1 Diodo 1N4000 ou similar
1 Transformador 12+12V/100mA
1 Eletrolítico 470µF/63 Volts
1 Capacitor 0,22µF de poliester
1 Capacitor 0,47µF de poliester
1 Resistor 4,7k Ω1/4 Watt
1 Resistor R_x (vide texto)

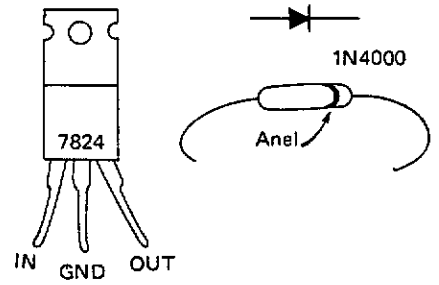


Fig. 27

Como não existe no mercado nacional um regulador de tensão de 25 Volts, utilizamos um artifício para extrair esta tensão a partir de um regulador de 24 Volts, facilmente encontrado na praça. O resistor Rx, ligado ao negativo (GND) do regulador, eleva a tensão de saída até a tensão desejada. No meu projeto, usei um resistor de 120 Ohms. No seu caso, este valor poderia variar ligeiramente, e o valor correto terá que ser encontrado por tentativas. Aumentando o valor de Rx, a tensão de saída cresce, e naturalmente, uma diminuição deste valor reflete-se numa tensão de saída menor.

Um programador de EPROM profissional emprega em geral um soquete de "zero-força", que apresenta contatos frouxos e uma alavanca, para apertar os pinos da EPROM a ser programada.

Como este soquete apresenta um custo bastante elevado, podemos montar o nosso programador com um soquete comum de boa qualidade. O único cuidado do operador é na hora de colocar e retirar a EPROM deste soquete, para não entortar ou até quebrar os pinos frágeis deste circuito integrado.

A ligação deste soquete com o circuito básico pode ser vista na Fig. 28, e a chave Ch será usada para selecionar entre o tipo 2716 e 2732. Temos ainda a chave Ch1 para ligar e desligar os 25 Volts.

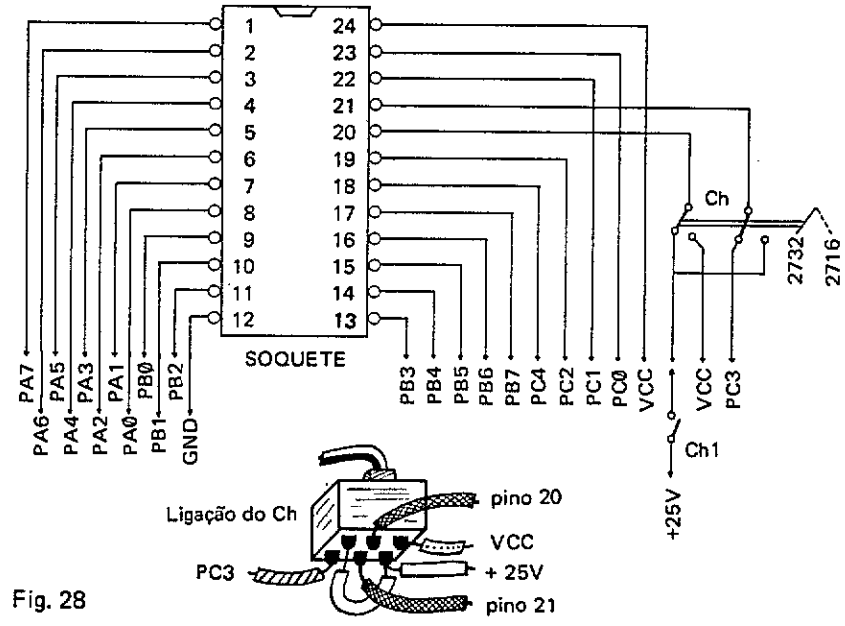


Fig. 28

Uma vez tudo ligado, faltam-nos ainda os programas correspondentes, divididos em linguagem de máquina e BASIC. O programa é auto-explicativo, e o único perigo em operar com este gravador de EPROM é nunca esquecer de deixar os 25 Volts desligados na hora de colocar ou de retirar a EPROM no soquete.

```
ROTINA ASSEMBLY 19    N=74


*16514 00 00 06 3E 30 32 03 243
 16521 20 C9 3E 00 32 02 20 379
*16528 C9 3E 10 16 F6 3A 82 739
 16535 40 32 00 20 3A 84 40 400
*16542 32 01 20 3A 83 40 C9 537
 16549 32 02 20 06 31 C5 06 342
*16556 FD 10 FE C1 10 F6 3A 1038
 16563 83 40 C9 CD 95 40 06 820
*16570 10 80 CD A5 40 18 CC 806
 16577 CD 95 40 CD A5 40 06 858
*16584 10 80 18 F3           411
```

```
ROTINA BASIC 30


   0 REM   (ROTINA ASSEMBLY 19)
  10 PRINT AT 5,4;"PROGRAMADOR D
E EPROM"
  20 RAND USR 16517
  30 PRINT AT 11,0;"SELECIONE O
TIPO E ACERTE A CHA-VE"
  40 INPUT A
  50 IF A<>2716 AND A<>2732 THEN
 GOTO 40
  60 PRINT AT 13,14;A
  70 LET R=2047
  80 LET PR=16566
  90 LET DEF=16523
 100 IF A=2716 THEN GOTO 140
 110 LET R=4095
 120 LET PR=16577
 130 LET DEF=16529
 140 RAND USR DEF
```



```
 150 PRINT AT 14,0;"LIGUE OS 25
VOLTS"
 160 PRINT AT 17,0;"QUER ENTRAR
MANUALMENTE (M) OU  COPIAR (C) U
MA MEMORIA ?"
 170 INPUT A$
 180 IF A$<>"M" AND A$<>"C" THEN
 GOTO 170
 190 IF A$="M" THEN GOTO 390
 200 CLS
 210 PRINT AT 5,0;"ENTRE COM O E
NDERECO INICIAL DA COPIA ";
 220 INPUT E
 230 PRINT E
 240 PRINT AT 7,0;"ENTRE COM O E
NDERECO INICIAL DA EPROM ";
 250 INPUT RO
 252 PRINT RO
 254 PRINT AT 9,0;"ENTRE COM O U
LTIMO ENDERECO ";
 256 INPUT R
 258 PRINT R
 260 PRINT AT 19,0;"APERTE N/L P
ARA INICIAR."
 270 INPUT A$
 275 FAST
 280 FOR F=RO TO R
 290 GOSUB 360
 300 POKE 16515,PEEK (E+F-RO)
 310 RAND USR PR
 320 NEXT F
 330 SLOW
 340 PRINT AT 21,0;"FIM. DESLIGU
E OS 25 VOLTS"
 350 STOP
 360 POKE 16514,F-INT (F/256)*25
6
 370 POKE 16515,INT (F/256)
 380 RETURN
 390 CLS
 400 PRINT AT 5,0;"ENDERECO INIC
IAL DA EPROM ?      ";
 410 INPUT F
 420 PRINT F;AT 7,0;"APERTE N/L
PARA INICIAR."
 430 INPUT A$
 440 CLS
 450 LET D$=""
 460 FAST
 465 SCROLL
 470 PRINT F;TAB 5;CHR$ (INT (E/
4096)+28);
```

```
  480 LET K=F-INT (F/4096)*4096
  490 PRINT CHR$ INT (K/256+28);
  500 LET K=K-INT (K/256)*256
  510 PRINT CHR$ INT (K/16+28);CH
R$ (K-INT (K/16)*16+28);" ";
  520 IF D$="" THEN INPUT D$
  530 PRINT D$(1 TO 2)
  540 POKE 16516,16*(CODE D$-28)+
CODE D$(2)-28
  550 GOSUB 360
  560 RAND USR PR
  570 LET F=F+1
  580 LET D$=D$(3 TO )
  590 GOTO 470
```

Agora você poderia, por exemplo, gravar definitivamente suas sub-rotinas favoritas, incluindo talvez algumas rotinas apresentadas neste livro.

SUGESTÕES

Espero que este livro tenha colaborado para criar novos projetos com o circuito básico, e incentivado os leitores a montar alguns dos circuitos apresentados.

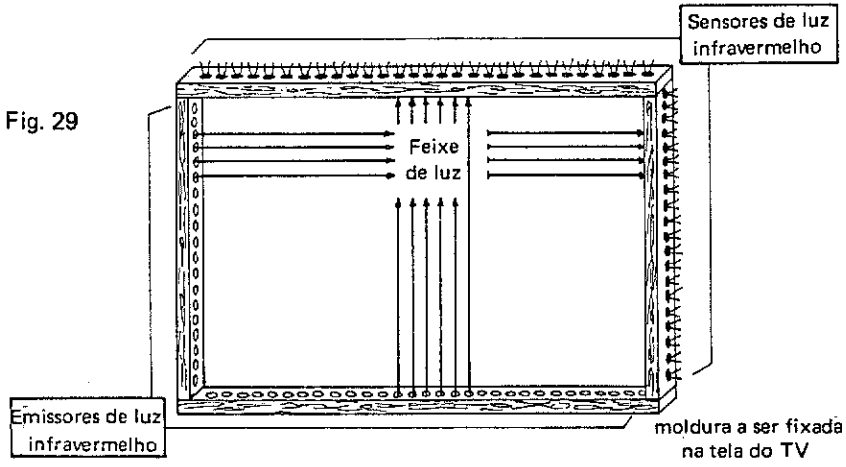
Como você pode reparar, com o circuito básico praticamente não há limites. A seguir há algumas sugestões, que talvez se tornem realidade nas suas mãos:

1. TELA SENSÍVEL AO TOQUE

Em computadores avançados, a seleção de um menu é feita tocando simplesmente a tela nos pontos indicados com o dedo, dispensando assim quase por completo o teclado. Isto não tem nada a ver com o monitor usado, e até o seu televisor pode ser usado com estes recursos.

Na verdade, existe uma matriz invisível de barreiras de luz infravermelha, onde a interrupção de qualquer um dos feixes é analisada pelo computador, reconhecendo assim a posição, onde houve este sinal (Fig. 29).

Fig. 29



Desta maneira, instalando uma barreira de luz infravermelha por exemplo na altura da linha 20 do computador, com o sensor conectado em A0, poderíamos por exemplo ter este trecho de um programa BASIC:

```
ROTINA BASIC 31

 130 PRINT AT 20,0;"XXXXXXXXXXXXX
XXXXXXXXXXXXXXXXXXX"
 140 PRINT AT 21,0;"TOQUE A LINH
A ACIMA PARA PARAR."
 150 IF PEEK 8192=1 THEN STOP
```

2. TECLADO INTELIGENTE

Podemos programar várias sub-rotinas no computador, que rodam conforme a configuração das entradas do circuito básico. Assim, ligando um teclado auxiliar de até 24 teclas (Fig. 30) ou de até 144 teclas (Fig. 31), podemos programar de tal modo, que cada tecla corresponda a uma rotina inteira. Deste modo, seria possível programar uma tecla para fazer um cálculo complicado e extenso, enquanto uma outra tecla grava o programa numa fita.

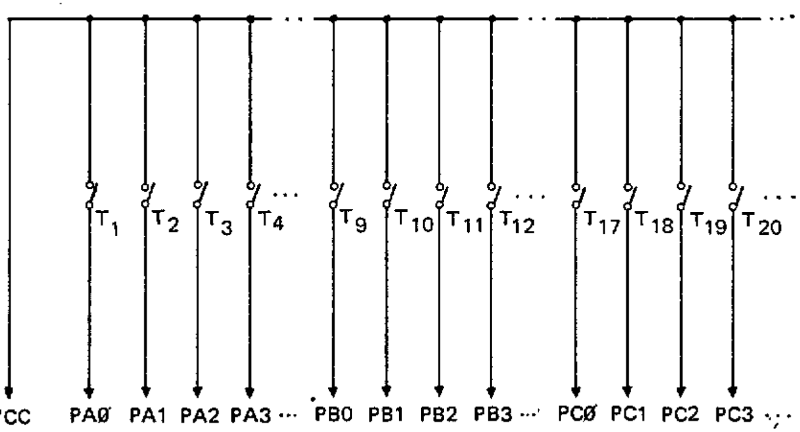


Fig. 30

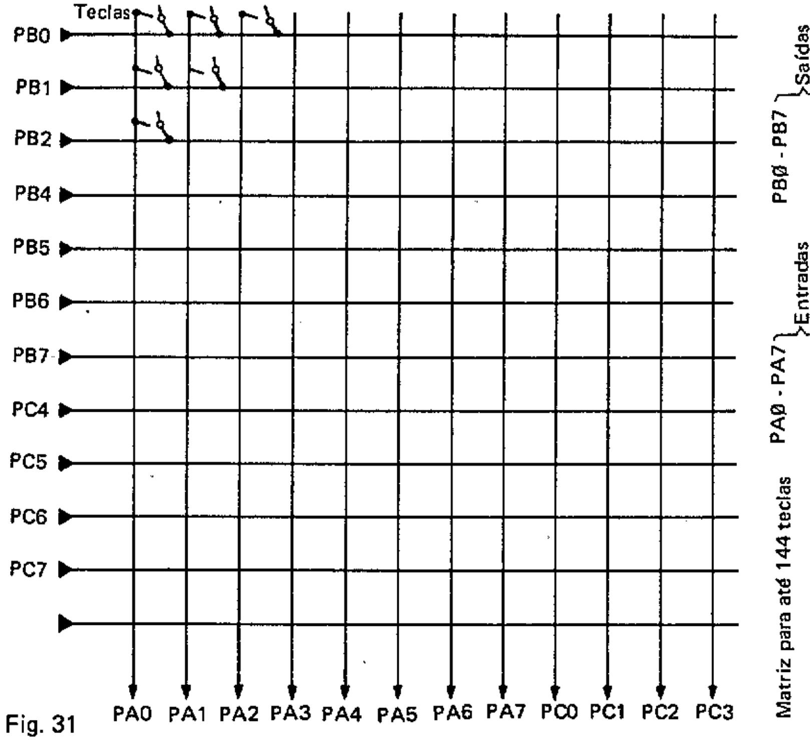


Fig. 31

3. COMANDO NUMÉRICO PARA MÁQUINAS

Por que não usar a capacidade do microcomputador e os recursos do circuito básico para comandar máquinas? Com auxílio dos relés, podemos transformar por exemplo um torno semi-automático numa máquina totalmente computadorizada, capaz de realizar processos de fabricação sem interferência de um operador. É só adicionar à máquina fins de curso, barreiras de luz e escrever a rotina de trabalho.

4. PX E PY

A maioria dos transmissores de PX e de PY usam um circuito integrado (PPL), para a seleção dos canais, onde a chave seletora transmite um código binário ou BCD para este PPL. Neste caso, a chave seletora pode ser substituída por nosso circuito básico, para realizar varreduras automáticas dos canais e outras aplicações conforme seu agrado.

5. ACIONAMENTO DE IMPRESSORAS E MÁQUINAS DE ESCREVER ELETRÔNICAS

Conhecendo os códigos para cada caractere numa impressora ou máquina de escrever eletrônica, é possível ligar seu micro a estes aparelhos, com ajuda de um pequeno programa "tradutor", que transforma os códigos dos caracteres do seu micro para os códigos do aparelho. É evidente, que com isto se perderá muito em velocidade, mas tenho certeza, que o resultado superará de longe esta deficiência.

6. EFEITOS SONOROS

Existem no mercado nacional vários sintetizadores de som, capazes de criar os mais incríveis sons, como por exemplo tiros, explosões, sirene e tons musicais.

Para a seleção dos diversos tons, existem em geral diversas entradas lógicas, que conforme o nível de entrada emitem um ou outro som. E, quem mais do que o nosso circuito básico para selecionar estes sintetizadores com simples POKE's?

A maioria destes circuitos integrados contém um oscilador controlando a voltagem (VCO), que conforme o nível DC emitem um tom com altura diferente. Para criar vários níveis DC com o circuito básico, podemos utilizar-nos do circuito da Fig. 32.

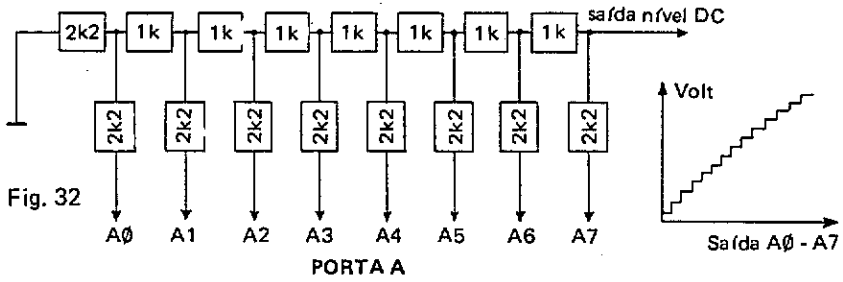


Fig. 32

Desta maneira, conforme o valor presente na Porta A, temos até 256 níveis DC diferentes, atingindo por exemplo no caso do sintetizador de som 76477 quase três oitavas.

7. Novos horizontes abrem-se usando o circuito básico para controlar o nível de líquidos, usando-o como freqüencímetro, e em conjunto com um conversor AD podemos montar multímetros, medidores de temperatura, e assim por diante.

TABELAS E DICAS

VALORES ÚTEIS PARA PROGRAMAR O 8255A:

POKE 8192,		Porta A	Porta B	Porta C	
decimal	hexadec.	PA0 até PA7	PB0 até PB7	PC0 - PC3	PC4 - PC
128	80	saída	saída	saída	saída
144	90	entrada	saída	saída	saída
130	82	saída	entrada	saída	saída
146	92	entrada	entrada	saída	saída
129	81	saída	saída	entrada	saída
145	91	entrada	saída	entrada	saída
131	83	saída	entrada	entrada	saída
147	93	entrada	entrada	entrada	saída
136	88	saída	saída	saída	entrada
152	98	entrada	saída	saída	entrada
138	8A	saída	entrada	saída	entrada
154	9A	entrada	entrada	saída	entrada
137	89	saída	saída	entrada	entrada
153	99	entrada	saída	entrada	entrada
139	8B	saída	entrada	entrada	entrada
155	9B	entrada	entrada	entrada	entrada

DICAS:

Se uma porta foi programada para o modo de saída, e quer-se saber o valor presente nas suas linhas, podemos obter este valor com uma simples leitura:

Exemplo: programar a porta A para o modo OUTPUT e colocar nas linhas PAØ até PA7 o valor 1Ø

```
POKE 8195,131   – programa a porta A
POKE 8192,1Ø    – acende os LED's 1 e 3 (bit 1 e bit 3)
PRINTPEEK8192   – confirma o valor 1Ø
```

ATENÇÃO:

Este teste não é válido para o endereço de programação 8195. Assim, quando estiver em dúvida, qual foi o último valor de programação, terá de repetir a programação.

Uma vez programado, o PIO permanece deste modo até que a alimentação for cortada, ou colocado um novo valor no endereço 8195.

A Tabela a seguir mostra, como podemos manipular os bits da porta C individualmente:

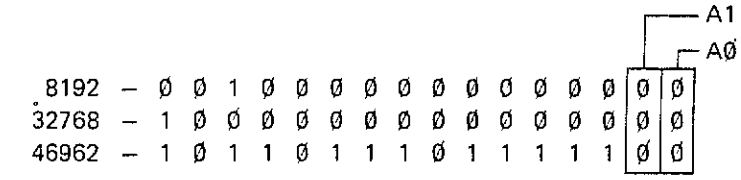
POKE 8195,				
bit 1 dec. / hex.		bit Ø dec. / hex.		Bit da porta C
1	Ø1	Ø	ØØ	Ø
3	Ø3	2	Ø2	1
5	Ø5	4	Ø4	2
7	Ø7	6	Ø6	3
9	Ø9	8	Ø8	4
11	ØB	1Ø	ØA	5
13	ØD	12	ØC	6
15	ØF	14	ØE	7

Endereços e operação do 8255A PIO:

Em princípio, o PIO pode trabalhar em quase qualquer endereço, mas devemos observar sempre o seguinte:

1. sempre serão usados 4 endereços seguidos ocupados pelo PIO
2. esses endereços não devem ser ocupados por outros periféricos (memórias RAM, EPROM, ROM etc.)

3. na configuração binária, o primeiro dos quatro endereços em questão, deve ter as linhas AØ e A1 com nível Ø.
   Exemplos para o primeiro endereço:



   e assim por diante

4. A grande vantagem de usar o PIO como uma memória, acessível com os comandos POKE e PEEK, é que podemos usar rotinas escritas em BASIC para trabalhar com este circuito integrado.
   Uma outra alternativa seria chavear o 8255A pelos ports disponíveis, dispensando assim a necessidade de reservar posições de memória.

   No caso do CP 2ØØ, alguns dos 256 ports disponíveis já são usados para o vídeo, teclado e gravador, e podemos usar alguns dos restantes para o circuito básico. Também aqui vale o critério do nível Ø das linhas AØ e A1 para o primeiro dos quatro ports necessários.

   Temos portanto teoricamente os seguintes ports disponíveis:

   ØØ, Ø4, Ø8, ØC, 1Ø, 14, 18, 1C, 2Ø, 24, 28, 2C, 3Ø etc.

   Resta eliminar os ports já usados pelo sistema, até encontrar 4 ports livres.

   Uma das desvantagens é o fato de que não teremos condições de acessar o 8255A somente com BASIC, e é preciso usar a linguagem de máquina através dos comandos IN e OUT.

   Para os usuários dos micros da linha TK, esse método não funciona sem alteração prévia do microcomputador, pois existe um "eco" dos ports usados em todos os demais ports.

ENDEREÇAMENTO INTERNO DO MICRO:

Há alguns casos particulares, onde a linha ROM CS não funciona para os endereços 8192 até 8195. Neste caso, é preciso alterar o sistema de endereçamento dentro do microcomputador.

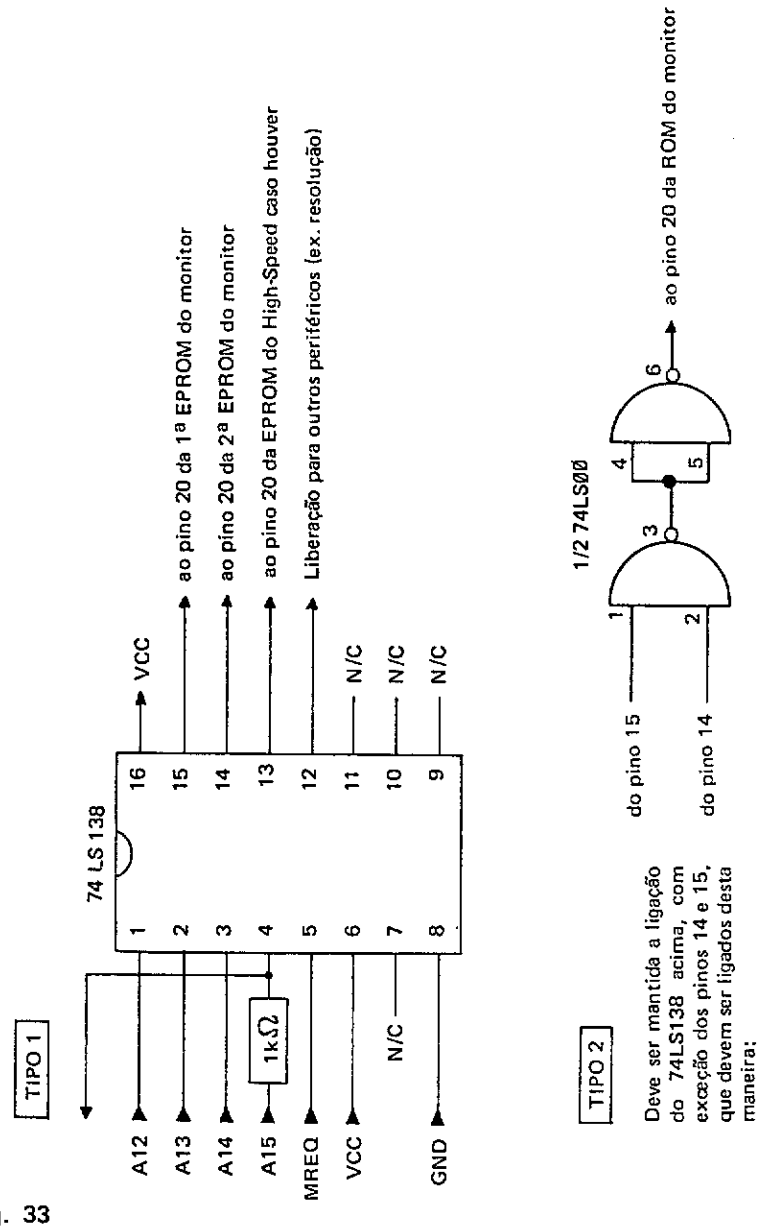


Fig. 33

Existem dois tipos de soluções, dependendo da forma como o monitor de 8 kBytes do seu micro é formado:

Tipo 1 — neste micro, o monitor é formado por duas EPROM's de 4 kBytes cada

Tipo 2 — neste outro, o monitor está contido numa única ROM de 8 kBytes

A alteração descrita a seguir não prejudica de maneira alguma o funcionamento normal do seu micro. Ao contrário, após a implantação desta modificação, será mais fácil incrementar seu micro com outros recursos, como alta resolução, High-Speed, sintetizador de som e de voz, etc. . .

Material necessário para micros do tipo 1:

1 resistor de 1 kohms 1/4 watt
1 circuito integrado 74LS138

Material necessário para micros do tipo 2:

1 resistor de 1 kohms 1/4 watt
1 circuito integrado 74LS138
1 circuito integrado 74LS00

A ligação destes componentes pode ser encontrada na Fig. 33, e deve ser realizada por um técnico.

Podemos ver que o chaveamento das ROM's é feito pelos pinos 20, que originalmente estavam interligados com o computador. É necessário isolar portanto estes pinos do computador e ligar nestes o 74LS138.

A linha ROM CS inhibe todas as memórias, colocando nesta linha um nível lógico 1.

AS ROTINAS ASSEMBLY DISASSEMBLADAS

Neste capítulo são relacionadas todas as rotinas em linguagem de máquina usadas neste livro, servindo de estudo para eventuais modificações ou transformações para o ASSEMBLY do seu computador.

```
Página 28          (Assembly 1)
16514  START       LD HL, (D—FILE)    2A 0C 40   ; localiza início da
                                                    tela
                   INC HL             23
                   LD (HL), 80        36 80      ; imprime CHR$ 80
                   RET                C9

Página 35          (Assembly 2)
16514  START       LD A, + 128        3E 80      ; programa a PIO
                   LD (8195), A       32 03 20
                   LD A, 0            3E 00      ; leva o valor de A
                   LD (8192), A       32 00 20     para a porta A
                   RET                C9

Página 36          (Assembly 3)
16514  START       JR CONT            18 0B      ; vai para o início
                   LD B, 0F           06 0F      ; sub rotina pausa
                   PUSH BC            C5
                   LD B, 0            06 00
                   DJNZ FE            10 FE
                   POP BC             C1
                   DJNZ F8            10 F8
                   RET                C9
```

```
        CONT    LD A, + 128     3E 80       ; programa a PIO
                LD (8195), A    32 03 20
                LD B, + 10      06 0A       ; define loop de 10
                PUSH BC         C5
                LD A, 1         3E 01       ; define o 1º LED
                LD B, 8         06 08       ; uma fase
                PUSH BC         C5            seqüencial
                LD (8192), A    32 00 20    ; acende o LED
                CALL PAUSA      CD 84 40
                RLA             17          ; passa p/próximo
                POP BC          C1            LED
                AND A           A7
                DJNZ F4         10 F4
                POP BC          C1
                DJNZ EC         10 EC
                LD A, 0         3E 00       ; apaga todos os
                LD (8192), A    32 00 20      LED's
                RET             C9
```

Página 37 (Assembly 4)

```
16514   START   JR CONT         18 0B       ; vai para o início
                LD B, + 15      06 0F       ; sub rotina de paus
                PUSH BC         C5
                LD B, 0         06 00
                DJNZ FE         10 FE
                POP BC          C1
                DJNZ F8         10 F8
                RET             C9
        CONT    LD A, + 128     3E 80       ; programa a PIO
                LD (8195), A    32 03 20
        NEXT    LD B, 0         06 00
                PUSH BC         C5
                LD A, 1         3E 01
                LD B, 8         06 08       ; define um loop
                PUSH BC         C5            completo
                LD (8192), A    32 00 20    ; acende o LED
                CALL PAUSE      CD 84 40
                RLA             17          ; passa para o
                POP BC          C1            próximo LED
```



```
                AND A           A7
                DJNZ F4         10 F4
                POP BC          C1
                DJNZ EC         10 EC
                LD A, 0         3E 00       ; apaga todos os
                LD (8192), A    32 00 20      LED's
                JR NEXT         18 E3
```

Página 39 (Assembly 5)

```
16514   START   JR CONT         18 0B       ; vai para o início
                LD B, + 15      06 0F       ; sub rotina de pausa
                PUSH BC         C5
                LD B, 0         06 00
                DJNZ FE         10 FE
                POP BC          C1
                DJNZ F8         10 F8
                RET             C9
        CONT    LD A, + 128     3E 80       ; programa a PIO
                LD (8195), A    32 03 20
                LD B, + 10      06 0A       ; define o loop de 10
                PUSH BC         C5
                LD A, 1         3E 01       ; define o 1º LED
                LD B, 8         06 08       ; sequência de 8
                PUSH BC         C5            LED's
                LD (8192), A    32 00 20    ; acende o LED
                CALL PAUSA      CD 84 40
                RLA             17          ; passa para o
                POP BC          C1            próximo LED
                DJNZ F5         10 F5
                LD B, 8         06 08       ; nova seqüência de
                PUSH BC         C5            8 LED's
                RRA             1F          ; muda de LED
                LD (8192), A    32 00 20    ; acende o LED
                CALL PAUSA      CD 84 40
                POP BC          C1
                AND A           A7
                DJNZ F4         10 F4
                POP BC          C1
                DJNZ DF         10 DF
```

```
         LD A, 0          3E 00      ; apaga todos os
         LD (8192), A     32 00 20     LED's
         RET              C9
```

Página 41

```
         (Assembly 6)
16514 START JR CONT       18 0B      ; vai para o início
         LD B, + 15       06 0F      ; sub rotina de pausa
         PUSH BC          C5
         LD B, 0          06 00
         DJNZ FE          10 FE
         POP BC           C1
         DJNZ F8          10 F8
         RET              C9
      CONT LD A, + 128    3E 80      ; programa a PIO
         LD (8195), A     32 03 20
         LD B, + 10       06 0A      ; define loop de 10
         PUSH BC          C5
         LD A, + 17       3E 11
         LD (8192), A     32 00 20
         CALL DELAY       CD 84 40
         LD A, + 34       3E 22
         LD (8192), A     32 00 20
         CALL PAUSA       CD 84 40
         LD A, + 68       3E 44
         LD (8192), A     32 00 20
         CALL PAUSA       CD 84 40
         LD A, + 136      3E 88
         LD (8192), A     32 00 20
         CALL PAUSA       CD 84 40
         POP BC           C1
         DJNZ DC          10 DC
         LD A, 0          3E 00      ; apaga todos os
         LD (8192), A     32 00 20     LED's
         RET              C9
```

Página 42

```
         (Assembly 7)
16514 START JR CONT       18 26      ; vai para o início
         LD B, + 15       06 0F      ; sub rotina de pausa
         PUSH BC          C5
```



```
         LD B, 0          06 00
         DJNZ FE          10 FE
         POP BC           C1
         DJNZ F8          10 F8
         RET              C9
         NOP's            00 00
         LD B, 5          06 05      ; sub rotina
         PUSH BC          C5
         LD A, (408F)     3A 8F 40   ; carrega em A o
         LD (8192), A     32 00 20     valor do endereço
         CALL PAUSA       CD 84 40     408F e acende os
                                       respectivos LED's
         LD A, (4090)     3A 90 40   ; carrega em A o
         LD (8192), A     32 00 20     valor do endereço
         CALL PAUSA       CD 84 40     4090 e acende os
         POP BC           C1           respectivos LED's
         DJNZ EA          10 EA
         RET              C9
CONT     LD A, + 128      3E 80      ; programa a PIO
         LD (8195), A     32 03 20
         LD B, + 10       06 0A      ; define 10 loop's
         PUSH BC          C5
         LD HL, 1         21 01 00   ; coloca no endereço
         LD (408F), HL    22 8F 40     408F o valor 1
         CALL SUBROTINA   CD 91 40
         LD HL, 0103      21 03 01   ; coloca no endereço
         LD (408F), HL    22 8F 40     408F o valor 0103h
         CALL SUBROTINA   CD 92 40
         LD HL, 0307      21 07 03   ; coloca no endereço
         LD (408F), HL    22 8F 40     408F o valor 0307h
         CALL SUBROTINA   CD 91 40
         LD HL, 070F      21 0F 07   ; coloca no endereço
         LD (408F), HL    22 8F 40     408F o valor
         CALL SUBROTINA   CD 91 40     070Fh
         LD HL, 0F1F      21 1F 0F   ; coloca no endereço
         LD (408F), HL    22 8F 40     408F o valor
         CALL SUBROTINA   CD 91 40     0F1Fh
         LD HL, 1F3F      21 3F 1F   ; coloca no endereço
         LD (408F), HL    22 8F 40     408F o valor
```

```
                CALL SUBROTINA  CD 91 40   1F3Fh
                LD HL, 3F7F     21 7F 3F   ; coloca no endereço
                LD (408F), HL   22 8F 40     408F o valor
                CALL SUBROTINA  CD 91 40   3F7Fh
                LD HL, 7FFF     21 FF 7F   ; coloca no endereço
                LD (408F), HL   22 8F 40     408F o valor
                CALL SUBROTINA  CD 91 40   7FFFh
                LD HL, 00FF     21 FF 00   ; coloca no endereço
                LD (408F), HL   22 8F 40     408F o valor
                CALL SUBROTINA  CD 91 40   00FFh
                POP BC          C1
                DJNZ AB         10 AB
                RET             C9
```

Página 46

```
                (Assembly 8)
16514  DADOS    NOP'S           00 00
                PUSH BC         C5         ; sub-rotina de pausa
                LD B, + 10      06 0A
                PUSH BC         C5
                LD B, 0         06 00
                DJNZ FE         10 FE
                POP BC          C1
                DJNZ F8         10 F8
                POP BC          C1
                RET             C9
       INICIO   LD A, + 137     3E 89      ; programa a PIO
                LD (8195), A    32 03 20
                LD A, 0         3E 00
                LD (4082), A    32 82 40   ; define os dados
                LD A, + 131     3E 83
                LD (4083), A    32 83 40
                LD B, 7         06 07
                LD A,1          3E 01
                LD (8192), A    32 00 20   ; acende um LED
                CALL PAUSA      CD 84 40
                PUSH AF         F5         ; guarda o valor de A
                LD A, (8194)    3A 02 20   ; lê a porta C
                CP 0            FE 00      ; verifica se tudo é 0
                JR NZ + 6       20 06
```



```
                POP AF          F1
                RLA             17         ; passa p/ o próximo
                DJNZ EE         10 EE        LED
                JR E8           18 E8
                POP AF          F1
                AND A           A7
                CP 8            FE 08      ; testa, se o LED 4 é
                JR NZ + 7       20 07        aceso
                LD A, (4083)    3A 83 40
                INC A           3C
                LD (4083), A    32 83 40
                LD A, (4082)    3A 82 40
                INC A           3C
                LD (4082), A    32 82 40
                AND A           A7
                CP + 10         FE 0A
                JR Z + 9        28 09
                LD A, (8194)    3A 02 20   ; lê a porta C
                CP 0            FE 00
                JR NZ F9        20 F9
                JR C6           18 C6
                LD BC, 0B08     01 08 0B   ; coloca a posição do
                CALL AT         CD F5 08     PRINT
                LD BC, + 16     01 10 00
                LD DE, 40 EE    11 EE 40
                CALL PRINT      CD 6B 0B   ; imprime a
                LD A, (4083)    3A 83 40     mensagem na tela
                ADD + 28        C6 1C
                RST 10          D7         ; imprime o número
                RET             C9
MENSAGEM        36 39 29 1B 29 2A 00 26 28
                2A 37 39 34 38 0E 00
```

Página 47

```
                (Assembly 9)
16514  DADOS    NOP's           00 00
                PUSH BC         C5         ; sub-rotina de pausa
                LD B, + 35      06 23
                PUSH BC         C5
                LD B, + 128     06 80
```

```
                DJNZ FE             10 FE
                POP BC              C1
                DJNZ F8             10 F8
                POP BC              C1
                RET                 C9
MENSAGEM        00 36 39 29 1B 00 39 2E
                37 34 38 0E 00 36 39 29
                1B 00 26 28 2A 37 39 34
                38 0E 00
                LD A, + 35          3E 23     ; define os dados
                LD (4086), A        32 86 40
                LD A, + 137         3E 89     ; programa a PIO
                LD (8195), A        32 03 20
                LD A, 0             3E 00
                LD (4082), A        32 82 40
                LD (4083), A        32 83 40
                LD A, 1             3E 01
                LD (4091), A        32 91 40
                LD B, 8             06 08
                PUSH BC             C5
                LD BC, 0109         01 09 01  ; define a posição do
                CALL AT             CD F5 08    PRINT
                LD BC, + 12         01 0C 00
                LD DE, 4092         11 92 40
                CALL PRINT          CD 6B 0B  ; imprime a
                LD A, (4082)        3A 82 40    mensagem
                CP + 10             FE 0A
                JR NZ + 7           20 07
                LD A, "1"           3E 1D
                RST 10              D7        ; imprime o número
                LD A, 0             3E 00       1
                NOP's               00 00
                ADD + 28            C6 1C
                RST 10              D7        ; imprime o número
                                                0
                LD BC, 0C09         01 09 0C  ; define a posição do
                CALL AT             CD F5 08    PRINT
                LD BC, + 14         01 0E 00
                LD DE, 409E         11 9E 40
```



```
                CALL PRINT          CD 6B 0B  ; imprime a
                LD A, (4083)        3A 83 40    mensagem
                CP + 10             FE 0A
                JR NZ + 7           20 07
                LD A, "1"           3E 1D
                RST 10              D7        ; imprime o número
                LD A, 0             3E 00       1
                NOP's               00 00
                ADD + 28            C6 1C
                RST 10              D7        ; imprime o número
                LD A, (4091)        3A 91 40    0
                LD (8192), A        32 00 20  ; acende os LED's
                RLA                 17
                LD (4091), A        32 91 40
                CALL PAUSA          CD 84 40
                LD A, (8194)        3A 02 20  ; lê a porta C
                CP 0                FE 00
                JR NZ + 5           20 05
                POP BC              C1
                DJNZ A8             10 A8
                JR 9F               18 9F
                POP BC              C1
                LD A, (4091)        3A 91 40
                CP + 16             FE 10
                JR Z + 22           28 16
                LD A, (4082)        3A 82 40
                INC A               3C
                LD (4082), A        32 82 40
                CP + 11             FE 0B
                RET Z               C8        ; retorna ao BASIC
                LD A, (8194)        3A 02 20    após 10 tiros
                CP 0                FE 00
                JR NZ F9            20 F9
                CALL PAUSA          CD 84 40
                JR E0               18 E0
                LD A, (4083)        3A 83 40
                INC A               3C
                LD (4083), A        32 83 40
                LD A, (4086)        3A 86 40
```

```
                AND A           A7
                SUB 3           D6 03       ; aumenta a
                LD (4086), A    32 86 40      velocidade
                JR D8           18 D8
```

Página 50

```
16514  START    (Assembly 10)
                LD A, + 130     3E 82       ; programa a PIO
                LD (8195), A    32 03 20
                LD A, (8193)    3A 01 20    ; lê a porta B
                LD (8192), A    32 00 20    ; e acende os
                JR F8           18 F8         respectivos LED's
```

Página 50

```
16514  DADO     (Assembly 11)
                NOP             00
                JR Z + 4        28 04       ; sub rotina, análise
                LD A, "1"       3E 1D         bit 0 ou bit 1
                RST 10          D7          ; imprime 1 na tela
                RET             C9
                LD A, "0"       3E 1C       ; imprime 0 na tela
                JR FA           18 FA
                LD A, + 155     3E 9B       ; programa a PIO
                LD (8195), A    32 03 20
                LD A, (8192)    3A 00 20    ; lê a porta A
                LD (4082), A    32 82 40
                LD BC, 0808     01 08 08    ; define a posição do
                CALL AT         CD F5 08      PRINT
                LD A, (4082)    3A 82 40
                BIT 7, A        CB 47       ; testa PA7
                CALL ANALISE    CD 83 40
                LD A, (4082)    3A 82 40
                BIT 6, A        CB 4F       ; testa PA6
                CALL ANALISE    CD 83 40
                LD A, (4082)    3A 82 40
                BIT 5, A        CB 57       ; testa PA5
                CALL ANALISE    CD 83 40
                LD A, (4082)    3A 82 40
                BIT 4, A        CB 5F       ; testa PA4
                CALL ANALISE    CD 83 40
                LD A, (4082)    3A 82 40
```



```
                BIT 3, A        CB 67       ; testa PA3
                CALL ANALISE    CD 83 40
                LD A, (4082)    3A 82 40
                BIT 2, A        CB 6F       ; testa PA2
                CALL ANALISE    CD 83 40
                LD A, (4082)    3A 82 40
                BIT 1, A        CB 77       ; testa PA1
                CALL ANALISE    CD 83 40
                LD A, (4082)    3A 82 40
                BIT 0, A        CB 7F       ; testa PAO
                JP ANALISE      C3 83 40    ; testa PAO
```

Página 51

```
16514  DADOS    (Assembly 12)
                NOP's           00 00 00
                JR Z + 4        28 04       ; sub rotina de
                LD A, "1"       3E 1D         análise
                RST 10          D7          ; imprime o númro 1
                RET             C9
                LD A, "0"       3E 1C       ; imprime o número
                JR FA           18 FA         0
                LD A, + 155     3E 9B       ; programa a PIO
                LD (8195), A    32 03 20
                LD A, (8192)    3A 00 20    ; lê a porta A
                LD (4082), A    32 82 40
                LD A, (8193)    3A 01 20    ; lê a porta B
                LD (4083), A    32 83 40
                LD A, (8194)    3A 02 20    ; lê a porta C
                LD (4084), A    32 84 40
                LD BC, 0A0F     01 0F 0A    ; define a posição do
                CALL AT         CD F5 08      PRINT
                LD A, (4082)    3A 82 40
                CALL SUBROTINA  CD CB 40
                LD BC, 0C0F     01 0F 0C    ; define a posição do
                CALL AT         CD F5 08      PRINT
                LD A, (4083)    3A 83 40
                CALL SUBROTINA  CD CB 40
                LD BC, 0E0F     01 0F 0E    ; define a posição do
                CALL AT         CD F5 08      PRINT
```

```
         LD A,(4084)       3A 84 40
         CALL SUBROTINA    CD CB 40
         RET               C9
SUBROTINA PUSH AF          F5
         BIT 7, A          CB 47     ; testa linha 7
         CALL ANALISE      CD 85 40
         POP AF            F1
         PUSH AF           F5
         BIT 6,A           CB 4F     ; testa linha 6
         CALL ANALISE      CD 85 40
         POP AF            F1
         PUSH AF           F5
         BIT 5, A          CB 57     ; testa linha 5
         CALL ANALISE      CD 85 40
         POP AF            F1
         PUSH AF           F5
         BIT 4, A          CB 5F     ; testa linha 4
         CALL ANALISE      CD 85 40
         POP AF            F1
         PUSH AF           F5
         BIT 3,A           CB 67     ; testa linha 3
         CALL ANALISE      CD 85 40
         POP AF            F1
         PUSH AF           F5
         BIT 2, A          CB 6F     ; testa linha 2
         CALL ANALISE      CD 85 40
         POP AF            F1
         PUSH AF           F5
         BIT 1, A          CB 77     ; testa linha 1
         CALL ANALISE      CD 85 40
         POP AF            F1
         BIT 0, A          CB 7F     ; testa linha 0
         CALL ANALISE      CD 85 40
         RET               C9
```

Página 53 (Assembly 13)

```
16514    Variável          00 00     ; para guardar
                                       DATAPOS
         Variável          00 00     ; para guardar
```



```
         LD HL, (DATAPOS)  22 82 40  ; começa a analisar
                                       POSTELA
         LD, HL, (VARS)    2A 10 40  ; rotina de medição
         LD BC, 5          01 05 00
         ADD HL, BC        09
         LD B,+ 28         06 1C     ; contra 28 medições
         PUSH BC           C5
         LD A, (8194)      3A 02 20  ; lê a porta C
         AND 80            E6 80     ; isola PC7
         LD B, A           47
         LD A, (8194)      3A 02 20  ; lê novamente a
         AND 80            E6 80       porta C e verifica,
         CP B              B8          se o nível mudou
         JR Z F8           28 F8     ; aguarde (opcional)
         LD A, (8192)      3A 00 20  ; lê a porta A
         LD (HL), A        77        ; guarda o resultado
         INC HL            23
         LD A, (8193)      3A 01 20  ; lê a porta B
         LD (HL), A        77        ; guarda o resultado
         INC HL            23
         LD A, (8194)      3A 02 20  ; lê a porta C
         LD (HL), A        77        ; guarda o resultado
         INC HL            23
         POP BC            C1
         DJNZ DF           10 DF     ; repita 28 vezes
         RET               C9
16561 ANÁLISE LD HL, (VARS) 2A 10 40  ; fixa DATAPOS
         LD BC, 5          01 05 00
         ADD HL, BC        09
         LD (DATAPOS), HL  22 82 40
         LD HL, (D-FILE)   2A 0C 40  ; define POSTELA
         LD BC, 5          01 05 00
         ADD HL, BC        09
         LD (POSTELA), HL  22 84 40
         LD B, + 28        06 1C
         PUSH BC           C5
         LD B, + 3         06 03
         PUSH BC           C5
         PUSH HL           E5
```

```
                LD A, (HL)          7E          as 28 medições
                INC HL              23
                LD (DATAPOS), HL    22 82 40
                POP HL              E1
                CALL PRINT BIT      CD E6 40    ; imprime "1" ou "0"
                POP BC              C1
                DJNZ EF             10 EF
                LD HL, (POSTELA)    2A 84 40
                INC HL              23
                LD (POSTELA), HL    22 84 40
                POP BC              C1
                DJNZ E2             10 E2
                RET                 C9          ; retorne ao BASIC
16614 PRINTBIT  BIT 0, A            CB 47       ; testa bit 0
                CALL 0/1            CD 0F 41
                BIT 1, A            CB 4F       ; testa bit 1
                CALL 0/1            CD 0F 41
                BIT 2,A             CB 57       ; testa bit 2
                CALL 0/1            CD 0F 41
                BIT 3, A            CB 5F       ; testa bit 3
                CALL 0/1            CDOF 41
                BIT 4, A            CB67        ; testa bit 4
                CALL 0/1            CD OF 41
                BIT 5, A            CB 6F       ; testa bit 5
                CALL 0/1            CD 0F 41
                BIT 6, A            CB 77       ; testa bit 6
                CALL 0/1            CD 0F 41
                BIT 7, A            CB 7F       ; testa bit 7
                CALL 0/1            CD 0F 41
                RET                 C9
16655 0/1       JR Z + 10           28 0A       ; imprime "0" ou
                PUSH AF             F5            "1" conforme o bit
                LD A, "1"           3E 1D       a ser testado
                LD (HL), A          77
                POP AF              F1
                LD BC, + 33         01 21 00
                ADD HL, BC          09
                RET                 C9
                PUSH AF             F5
```



```
                LD A, "0"           3E 1C
                JR F4               18 F4

Página 55                           (Assembly 14)
16514 Variáveis DATAPOS             00 00
                POSTELA             00 00
16518 MEDIÇÃO   LD HL, (VARS)       2A 10 40    ; define DATAPOS
                LD BC, + 5          01 05 00
                ADD HL, BC          09
                LD (DATAPOS), HL    22 82 40
                LD B, + 28          06 1C       ; são 28 medições
                PUSH BC             C5
                LD A, (8193)        3A 01 20    ; lê a porta B
                AND 08              E6 08       ; isola PB3
                LD B, A             47
                LD A, (8193)        3A 01 20
                AND 08              E6 08
                CP B                B8          ; testa o "clock"
16543           JR Z F8             28 F8         (opcional)
                LD A, (8192)        3A 00 20    ; lê a porta A
                LD (HL), A          77
                INC HL              23
                LD A, (8193)        3A 01 20    ; lê a porta B
                LD (HL), A          77
                INC HL              23
                POP BC              C1
                DJNZ E4             10 E4       ; repete 28 vezes
                RET                 C9
16559 ANÁLISE   LD HL, (D-FILE)     2A 0C 40    ; define POSTELA
                LD BC, + 5          01 05 00
                ADD HL, BC          09
                LD (POSTELA), HL    22 84 40
                LD B, + 28          06 1C       ; são 28 resultados
                PUSH BC             C5
                PUSH HL             E5
                LD HL, (DATAPOS)    2A 82 40    ; pega um resultado
                LD A, (HL)          7E
                INC HL              23
                LD (DATAPOS), HL    22 82 40
```

```
                POP HL              E1
                CALL BIT 0-3        CD E4 40  ; imprime bit 0 até 3
                CALL BIT 4-7        CD F9 40  ; imprime bit 4 até 7
                PUSH HL             E5
                LD HL, (DATAPOS)    2A 82 40
                LD A, (HL)          7E
                INC HL              23
                LD (DATAPOS), HL    22 82 40
                POP HL              E1
                CALL BIT 0-3        CD E4 40  ; imprime PB0 até
                LD HL, (POSTELA)    2A 84 40    PB3
                INC HL              23
                LD (POSTELA), HL    22 84 40
                POP BC              C1
                DJNZ D8             10 D8
                RET                 C9
16612  BIT 0-3  BIT 0, A            CB 47     ; testa bit 0
                CALL PRINT          CD 0E 41
                BIT 1, A            CB 4F     ; testa bit 1
                CALL PRINT          CD 0E 41
                BIT 2, A            CB 57     ; testa bit 2
                CALL PRINT          CD 0E 41
                BIT 3, A            CB 5F     ; testa bit 3
                CALL PRINT          CD 0E 41
                RET                 C9
16633  BIT 4-7  BIT 4, A            CB 67     ; testa bit 4
                CALL PRINT          CD 0E 41
                BIT 5, A            CB 6F     ; testa bit 5
                CALL PRINT          CD 0E 41
                BIT 6, A            CB 77     ; testa bit 6
                CALL PRINT          CD 0E 41
                BIT 7, A            CB 7F     ; testa bit 7
                CALL PRINT          CD 0E 41
                RET                 C9
16654  PRINT    JR Z + 11           28 0B
                PUSH AF             F5
                LD A, "1"           3E 03     ; símbolo gráfico
                LD (HL), A          77          para "1"
                POP AF              F1
```



```
                LD BC, +66          01 42 00
                ADD HL, BC          09
                AND A               A7
                RET                 C9
                PUSH AF             F5
                LD A, "0"           3E 83     ; símbolo gráfico
                JR F3               18 F3       para "0"
```

Página 65 (Assembly 15)

```
16514  START    JR CONT             18 0E     ; vai para o início
                NOP's               00 00
                LD HL, (4084)       2A 84 40
                INC HL              23
                LD B, +32           06 20
                DJNZ FE             10 FE     ; atraso da rotina
                LD (4084), HL       22 84 40
                RET                 C9
         CONT   LD HL, 0            21 00 00  ; define as variáveis
                LD (4084), HL       22 84 40
                LD A, + 137         3E 89     ; programa a PIO
                LD (8195), A        32 03 20
                LD A, (8194)        3A 02 20  ; lê a porta C
                CP 0                FE 00
                JR NZ F9            20 F9
                LD A, +255          3E FF     ; acende todos os
                LD (8192), A        32 00 20    LED's
                LD A, (8194)        3A 02 20  ; lê a porta C
                CP 0                FE 00
                JR NZ + 5           20 05
                CALL ATRASO         CD 86 40
                JR F4               18 F4
                LD BC, (4084)       ED 4B 84 40
                LD A, 0             3E 00     ; apaga todos os
                LD (8192), A        32 00 20    LED's
                RET                 C9
```

Página 67 (Assembly 16)

```
16514  START    LD A, +130          3E 82     ; programa a PIO
                LD (8195), A        32 03 20
```

```
        LD HL, 0          21 00 00
        LD A, (8193)      3A 01 20   ; lê a porta B
        CP 0              FE 00
        JR NZ F9          20 F9
        LD A, 1           3E 01      ; liga o relé em PC0
        LD (8194), A      32 02 20
        LD A, (8193)      3A 01 20   ; lê a porta B
        CP 0              FE 00
        JR NZ OK          20 07
        INC HL            23         ; conta os
        LD B, +32         06 20        milissegundos
        DJNZ FE           10 FE
        JR F2             18 F2
OK      LD B, H           44         ; transfere o
        LD C, L           4D           resultado de HL
                                       para BC
        LD A, 0           3E 00      ; desliga o relé
        LD (8194), A      32 02 20
        RET               C9
```

Página 69

```
16514 START (Assembly 17)
        CALL FAST         CD 23 0F
        CALL FAST         CD 23 0F
        LD A, +128        3E 80      ; programa a PIO
        LD (8195), A      32 03 20
        LD A, 1           3E 01      ; liga o relé
        LD (8192), A      32 00 20
        LD B, 0           06 00      ; tempo para a fita
        PUSH BC           C5           ganhar a sua
        LD BC, 0          06 00        velocidade
        DJNZ FE           10 FE
        POP BC            C1
        DJNZ F8           10 F8
        CALL SAVE         CD 0B 03   ; grava o programa
        LD A, 0           3E 00      ; desliga o gravador
        LD (8192), A      32 00 20
        CALL SLOW         CD 2B 0F
        RET               C9
```


Página 69

```
16514 START (Assembly 18)
        CALL FAST         CD 23 0F
        LD A, +128        3E 80      ; programa a PIO
        LD (8195), A      32 03 20
        LD A, 1           3E 01      ; liga o gravador
        LD (8192), A      32 00 20
        CALL LOAD         CD 44 03   ; lê o programa da
                                       fita
        LD A, 0           3E 00      ; desliga o gravador
        LD (8192), A      32 00 20
        CALL SLOW         CD 2B 0F
        RET               C9
```

Página 88

```
16514 DADOS (Assembly 19)
        NOP's             00 00 00
        LD A, +128        3E 80      ; programa a PIO
        LD (8195), A      32 03 20
        RET               C9
        LD A, 0           3E 00      ; prepara para 2716
        LD (8194), A      32 02 20
        RET               C9
        LD A, +16         3E 10      ; prepara para 2732
        JR F8             18 F8
        LD A, (4082)      3A 82 40
        LD (8192), A      32 00 20   ; define PA0 até
        LD A, (4084)      3A 84 40     PA7
        LD (8193), A      32 01 20   ; define PB0 até PB7
        LD A, (4083)      3A 83 40
        RET               C9
        LD (8194), A      32 02 20   ; define porta C
        LD B, 31          06 31      ; tempo de 50 ms
        PUSH BC           C5
        LD B, FD          06 FD
        DJNZ FE           10 FE
        POP BC            C1
        DJNZ F8           10 F8
        LD A, (4083)      3A 83 40
        RET               C9
        CALL 4095         CD 95 40
```

```
LD B, +16          06 10      ; programação da
ADD B              80           EPROM
CALL TEMPO         CD A5 40
JR CC              18 CC
CALL 4095          CD 95 40
CALL TEMPO         CD A5 40
LD B, +16          06 10
ADDB               80
JR F3              18 F3
```


CONCLUSÃO

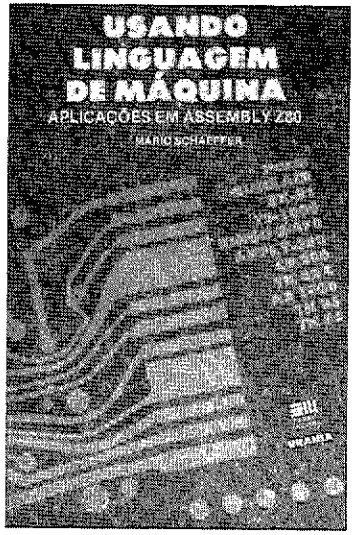
Todos os projetos apresentados neste livro foram montados pelo autor e os protótipos funcionaram perfeitamente. Naturalmente, todas as montagens podem ser aperfeiçoados por você, dependendo da sua habilidade.

O que realmente me levou a escrever este livro, é a escassez de literatura voltada para o "hardware" de microcomputadores, numa linguagem fácil e compreensível. São exigidos poucos conhecimentos para montar estes projetos, e com ajuda do circuito básico e os módulos apresentados como os relés, sensores etc., foi criado o primeiro degrau para uma nova ocupação do "hobbista": interessante, educativa e recompensável.

Meus agradecimentos às firmas e pessoas abaixo relacionadas, as quais me ajudaram muito a realizar esta obra:

- À minha esposa Leila e aos meus filhos Michaela e Michael pela paciência demonstrada durante a preparação desta obra.
- À Micromega/São Paulo pelo apoio e pela orientação prestada.
- Ao Sr. Dárcio Alves da Silva, pelas fotos publicadas neste livro.

coleção URANIA



USANDO LINGUAGEM DE MÁQUINA

APLICAÇÕES EM ASSEMBLY Z80

MÁRIO SCHAEFFER

Uma obra original, realmente didática, para aprendizado e consulta.

Instruções para o uso de linguagem de máquina em computadores compatíveis com Sinclair (RINGO, ZX-81, TS-1000, NEZ 8000, TK-82C, TK-85, TK-83).

Como usar as sub-rotinas da ROM, inclusive para cálculos científicos.

Mário Shaeffer mostra com muita inteligência e didática, como fazer verdadeiros milagres de programação utilizando Linguagem de Máquina, fornecendo muitíssimos exemplos de aplicação e brindando o leitor com uma série de programas úteis e divertidos.

DISSECANDO JOGOS

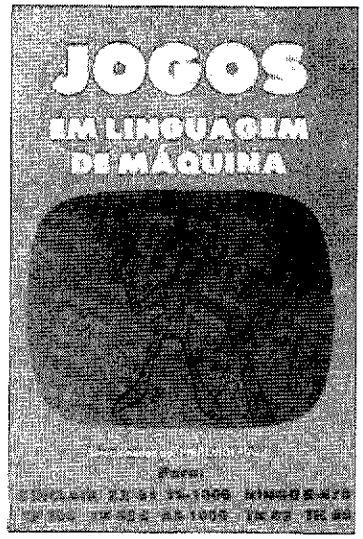
em BASIC TK comentado linha por linha

CARLOS EDUARDO ROCHA SALVATO

Os sete jogos contidos neste volume tem uma característica única: o leitor participa do processo de criação acompanhando a montagem linha por linha.

Modificações e aperfeiçoamentos são propostos e resolvidos com detalhes. Ao terminar este livro o leitor além de ter sete jogos geniais, terá adquirido a habilidade de programar com criatividade.

Uma obra didática indispensável ao usuário de um micro da linha SINCLAIR: Ringo R-470, CP-200, TK-82/83/85, AS-1000, SINCLAIR ZX-81 ou TS-1000.



JOGOS EM LINGUAGEM DE MÁQUINA - VOL I

Selecionados por PIERLUIGI PIAZZI

Use suas habilidades, sua inteligência e seus reflexos: viva os jogos que você mesmo digitou.

JOGOS EM LINGUAGEM DE MÁQUINA é um livro fascinante, rico em explicações do qual você extrairá uma quantidade enorme de programas.

A digitação dos programas não exige conhecimento de linguagem de máquina.

JOGOS EM LINGUAGEM DE MÁQUINA - VOL. II

Selecionados por PIERLUIGI PIAZZI

Para quem já se deliciou com o Volume I desta coleção, mais programas e jogos interessantíssimos e fascinantes:

JOGOS DE AÇÃO:

- WARDOZ
- CICLO TRON
- PAC MAN
- CORRIDA DO OURO
- TÚNEL

JOGOS INTELIGENTES: Criatividade e um magnífico xadrez, com instruções detalhadas e informações curiosas sobre este jogo tão antigo e tão atual.

Todos os jogos em ASSEMBLY Z-80, de execução rápida. Uma obra de lazer e de consulta para usuários de micro-computadores compatíveis com SINCLAIR ZX-81: TK-82, RINGO, R-470, CP-200, TK-83, AS-1000 e TK-85.

Pelo preço de uma fita, uma quantidade incrível de programas geniais!

JOGOS EM LINGUAGEM DE MÁQUINA - VOL. III

Selecionados por PIERLUIGI PIAZZI

Continuando a série, mais jogos inéditos:

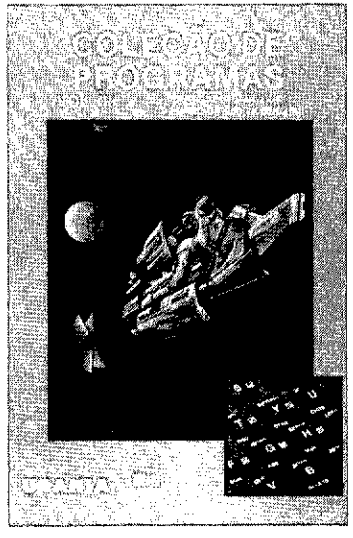
- INVASÃO
- I.R.A.
- VERMES DE AREIA (Duna)
- BASQUETE
- FROGGIE

e muitos outros.

Desenvolva o lado direito de seu cérebro (visão espacial) jogando o fascinante DÉDALO.

A digitação não exige conhecimento de Linguagem de Máquina.

Jogos em ASSEMBLY Z-80 para computadores compatíveis com SINCLAIR ZX-81 (TK-82/83/85, CP-200, TS-1000, etc.).

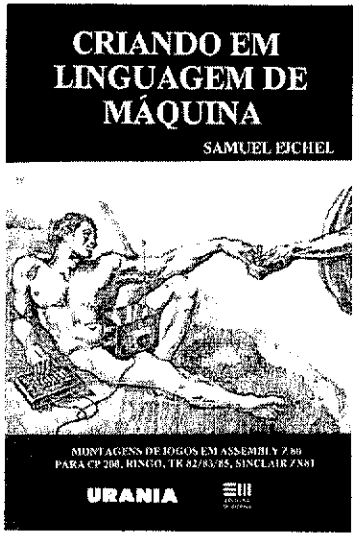


COLEÇÃO DE PROGRAMAS VOLUME III

RICARDO DE FAYETTI SIQUEIRA

Aos iniciantes no uso de um microcomputador pessoal, nada melhor, para o aprendizado, do que a digitação de programas e sua posterior análise e adaptação. Esta coleção pretende fornecer subsídios para o aprendizado e o lazer do usuário, apresentando tanto programas de jogos, quanto programas didáticos na área de matemática, da física e da química.

O Volume III da COLEÇÃO DE PROGRAMAS é portanto indicado para estudantes, adolescentes e iniciantes na fascinante técnica (e arte) de programar um computador pessoal.



CRIANDO EM LINGUAGEM DE MÁQUINA

SAMUEL EJCHEL

Não basta se conhecer as instruções do ASSEMBLY para se criar um programa interessante: existem truques e "dicas" indispensáveis para o bom programador.

Gerenciando a tela
Utilizando o teclado
Usando o Joystick
Gerando movimentos múltiplos
Criando opções
Implementando "requintes"

Em cada capítulo um jogo original, totalmente explicado, ilustrando os truques de programação utilizados.

Uma obra indispensáveis para quem programa em micros da linha SINCLAIR: CP-200 – RINGO – TK-82/83/85 – AS-1000.

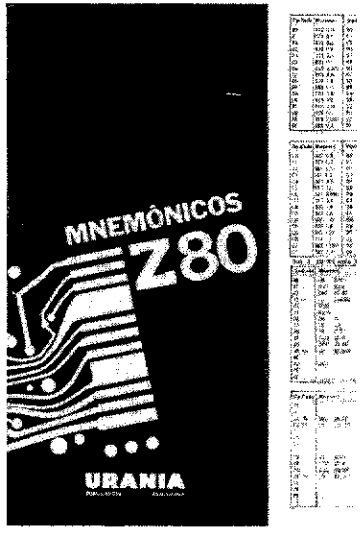


TABELA DE MNEMÔNICOS Z80

Tabela de cartolina plastificada para consulta rápida das instruções do Z80, seus códigos hexadecimais e abreviações mnemônicas. Indispensável para quem programa em linguagem de máquina.

Para computadores da linha SINCLAIR ZX81 (TK-82/83/85, CP-200, Ringo, AS-1000) e TRS80 (CP-300/500, JR SYSDATA, D-8000/8001/8002, DGT-100/101).

BASIC TK - VOL. II

PIERLUIGI PIAZZI/FLAVIO ROSSINI

A continuação de um dos maiores sucessos editoriais em informática.

Aprofundamento do BASIC com estrutura de programação.

Truques utilizando o sistema operacional dos micros da linha SINCLAIR (ZX-81, TK-82/83/85, CP-200, RINGO).

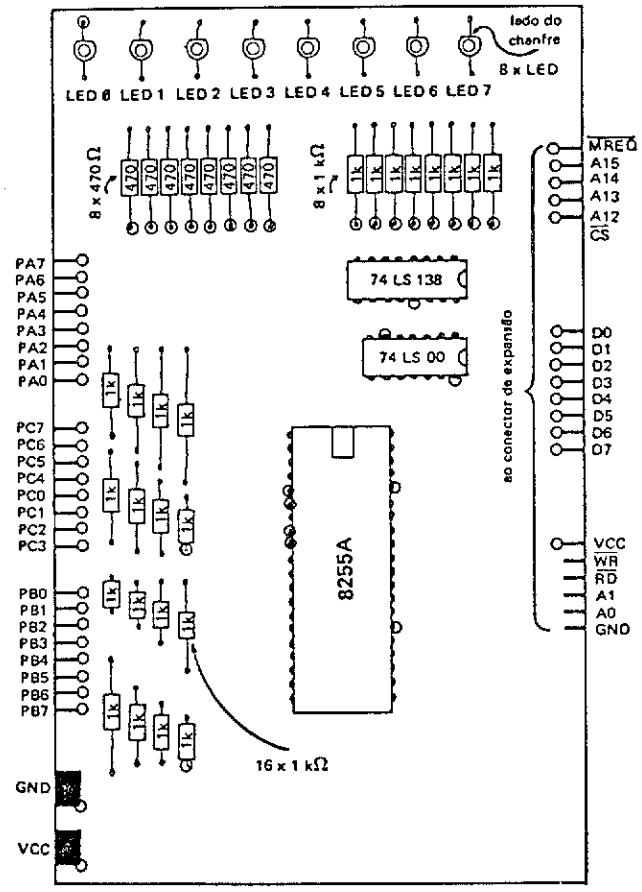
Uma obra elaborada com os mesmos cuidados didáticos do Volume I.

Querendo adquirir um destes livros escreva para:
ALEPH
PUBLICAÇÕES E ASSESSORIA PEDAGÓGICA LTDA.
Av. Brig. Faria Lima, 1451 - conj. 31
01451 - São Paulo - SP - Tel.: (011) 813-4555

Impresso na
planimpress gráfica e editora
rua anhaia, 247 - s.p.

Página 22

Fig. 7

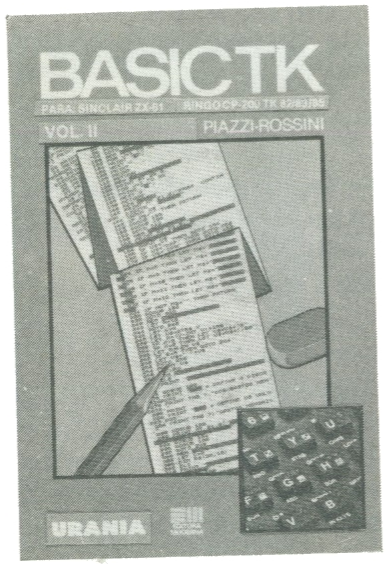


CIRCUITO BÁSICO
O SEU MICRO E O MUNDO EXTERNO

OS CIRCULOS INDICAM OS PONTOS ONDE O COMPONENTE DEVE SER SOLDADO NAS DUAS FACES DO CIRCUITO

NRO USAR A LIGACAO CS

ERRATA



BASIC TK - VOL. II

PIERLUIGI PIAZZI/FLAVIO ROSSINI

A continuação de um dos maiores sucessos editoriais em informática.

Aprofundamento do BASIC com estrutura de programação.

Truques utilizando o sistema operacional dos micros da linha SINCLAIR (ZX-81, TK-82/83/85, CP-200, RINGO).

Uma obra elaborada com os mesmos cuidados didáticos do Volume I.

Querendo adquirir um destes livros escreva para:

ALEPH

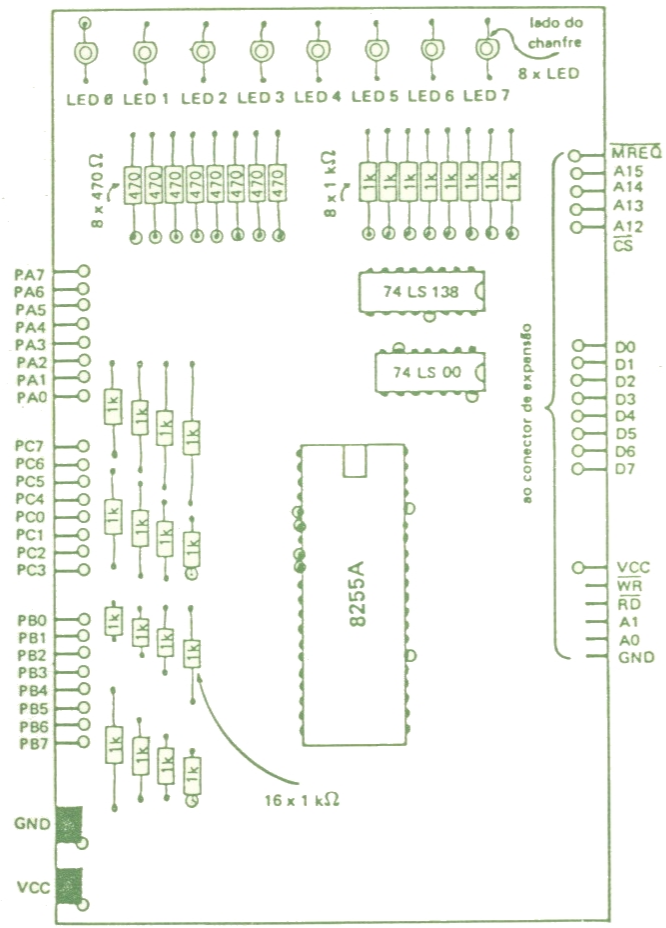
PUBLICAÇÕES E ASSESSORIA PEDAGÓGICA LTDA.

Av. Brig. Faria Lima, 1451 - conj. 31

01451 - São Paulo - SP - Tel.: (011) 813-4555

impresso na
planimpress gráfica e editora
rua anhaia, 247 - s.p.

Fig. 7



CIRCUITO BÁSICO
O SEU MICRO E O MUNDO EXTERNO

OS CIRCULOS INDICAM OS PONTOS ONDE O COMPONENTE DEVE SER SOLDADO NAS DUAS FACES DO CIRCUITO

NÃO USAR A LIGACAO CS

ERRATA