

```
*****
***** GIG DA PLACA DE COMUNICACAO DISCADA/VTX *****
***** I-7020 *****
*****
** objetivo : testar e ajustar os varios blocos (interfa- **
** ce de linha, controle digital e interface de **
** video) da placa de expansao I-7020. **
**
** testes : 1) Acesso ao lucy (SAA5070) **
**          2) Circuito de discagem **
**          3) Contoladores de video ; memoria de tela **
**          4) Transmissao/Recepcao (Ajuste) ; Som **
**
*****
```

```

;
;EQUATES
```

```

;
rcoma equ 52h
rcomb equ 53h
rdada equ 50h
rdadb equ 51h
loadm equ
loadm equ 80h
loady equ 20h
loadx equ 40h
loadr equ 0a0h
loady0 equ 0c0h
bgrow equ 00h
rcomm equ 63h
rmode equ 62h
rlthr equ 64h
porta equ 66h
portb equ 67h
rdial equ 68h
rsta0 equ 60h
```

```

;*****
;***** TESTE 1 : ACESSO AO LUCY *****
;*****
;* Escreve e le o registrador de comando (rcomm) verifican-*
;* se a leitura e correta.Se houver sucesso vai para o pro-*
;* ximo teste.Caso contrario entra em loop escrevendo 55 no *
;* registrador rcomm. *
;*****
.z80
test1:  ld a,55h          ;escreve 55h em rcomm
        out (rcomm),a
        in a,(rcomm)      ;le rcomm e compara com 55h
        cp 55h
        jp z,test2
loop1:  ld a,55h
        out (rcomm),a     ;entra em loop escrevendo 55h em rcomm
        jp loop1

```

```

; *****
; ***** TESTE 2 : DISCAGEM *****
; *****
; * aciona o circuito de discagem para verificacao de funcio- *
; * namento, discando 0's por tres vezes. *
; *****
test2:  ld a,0f4h          ;permite discagem; inicializa port A
        out (porta),a
        ld b,0fdh        ;inicializa port B
        out (portb),a
        ld c,03h         ;disca 0 por tres vezes
loop2:  ld a,0eah
        out (rdial),a
loop3:  in a,(rdial)       ;verifica se acabou a discagem
        bit 5,a
        jr nz,loop3
        dec c
        jr nz,loop2

```

```

; ****
; ***** TESTE 3 : controlador de video/memoria de tela *****
; ****
; * testa o acesso aos controladores de video , entrando em *
; * loop caso exista algum problema de escrita/leitura. Caso *
; * contrario escreve na memoria de video um padrao de barras *
; * coloridas. *
; ****
; test3:
    call buzzy                ;verifica se pode escrever
    ld a,85h                 ;programa display and timing mode register
    out (rcoma),a
    ld a,loadr
    out (rcomb),a
    ld b,00h                 ;no zoom mode e origem linha 00
    ld c,loady0
    call outcom
    ld b,00h                 ;begin row
    ld c,bgrow
    call outcom
    ld a,0ffh                ;chaveia sinais de video
    out (portb),a
    ld e,7fh                 ;E = caracter cheio semi-grafico
    ld hl,0000h              ;inicializa linha(H=0) e coluna(L=0)
    ld b,1                   ;B = coluna inicial da barra
    ld d,8fh                 ;D = atributo cor branca
    call esbar                ;escreve coluna branca
    ld d,8bh                 ;D = atributo cor amarela
    call esbar                ;escreve coluna amarela
    ld d,8eh                 ;D = atributo cor cyan
    call esbar                ;escreve coluna cyan
    ld d,8ah                 ;D = atributo cor verde
    call esbar                ;escreve coluna verde
    ld d,8dh                 ;D = atributo cor magenta
    call esbar                ;escreve coluna magenta
    ld d,89h                 ;D = atributo cor vermelho
    call esbar                ;escreve coluna vermelha
    ld d,8ch                 ;D = atributo cor azul
    call esbar                ;escreve coluna azul
    ld d,88h                 ;D = atributo cor preto
    call esbar                ;escreve coluna preta
    jp test4
esbar:  ld c,05h              ;escreve 5 colunas em todas as linhas
loop4:  call escar            ;a partir de uma coluna inicial em L
        inc l
        dec c
        jr nz,loop4
        ld
        ld a,h
        cp 21                ;verifica se chegou ao fim da coluna
        jr z,zesbar
        inc h
        ld l,b
        jp esbar
zesbar: xor a                ;inicializa linha
        ld h,a
        ld b,1                ;inicializa nova coluna
        ret

```

```

escar:  push bc          ;salva registradores
        ld b,h
        ld c,loady      ;posiciona linha do cursor
        call outcom
        ld b,l
        ld c,loadx      ;posiciona coluna do cursor
        call outcom
        ld b,40h        ;write without inc
        ld c,loadm
        call outcom
        ld b,d          ;BC = atributo+caracter
        ld c,e
        call outdad     ;escreve na memoria
        pop bc          ;restaura registradores
        ret

outcom:  call buzy       ;verifica se pode escrever
        ld a,b
        out (rcoma),a
        ld a,c
        out (rcomb),a
        ret

outdad:  call buzy       ;verif
        out (rcomb),a
        ret

outdad:  call buzy       ;verifica se pode escrever
        ld a,b
        out (rdada),a
        ld a,c
        out (rdadb),a
        ret

buzy:    in a,(rcoma)    ;verifica se pode escrever
        bit 7,a
        jr nz,buzy      ;se nao ,testa novamente
        ret

```

```

***** TESTE 4 : Transmissao/Ajuste de nivel/som *****
*****
** Fica em loop transmitindo caracteres 'U' com paridade par **
** (padrao de dados 010101...); Ajustar**
**
** 1) P101(50k), de modo que se obtenha um sinal de 1.0Vpp **
** (-7dBm) no catodo de D112 (1N751); **
** 2) P100(4k7), de modo a obter uma forma de onda simetrica **
** em RXDATA (pino 7 do LUCY); **
** 3) P102(4k7), de modo a obter no alto-falante (J4) um **
** volume satisfatorio. **
*****
test4:  ld a,0fch          ;inicializa port A , habilita som
        out (porta),a
        ld a,0f3h          ;inicializa port B ; TX enable
        out (portb),a
        ld a,20h           ;mode = transmissao 1200 bps
        out (rmode),a
        ld a,43h           ;command = ltxen
        out (rcomm),a
loop5:  in a,(resta0)       ;verifica se pode transmitir
        bit 7,a
        jr z,loop5

        ld a,55h           ;line transmit holding register = 55h
        out (rlthr),a
        jp loop5
end

```