# SVGA GENERATOR

**Roy Harding's VGA and SVGA test generator uses a PIC variant that's so fast that it is capable of producing the test signal without any external timing circuitry, making the hardware side easy to implement.**

As an electronics design engineer in the computer industry, I often get asked by friends and relatives to check or repair computers and monitors. If someone arrives with just a monitor for checking this involves wiring up a PC with a keyboard and mouse, then loading software and waiting while the whole thing boots-up.

The addition of a computer also takes up valuable workspace if one is working in a confined area. There are a few SVGA generators already available on the market, but the cost of these can only be justified if you are doing servicing or repairs full time, rather than as a favour.

Last year I was introduced to a new microcontroller from Scenix. This micro uses a combination of PIC code with additional instructions to enhance performance.

The device can operate up to 50MHz. It has a turbo setting to cancel the clock's internal divide-by-four, as found in PICs. This gives you 50MIPS performance, coupled with 2048 bytes of E2 flash memory, an eight-level stack and easy page switching. There's also 30mA source/sink capability on the I/O pins, which makes the device very useful.

As this micro works so fast, I thought of the idea of using it to create a video generator that could work up to high resolutions. The generator described will generate standard VGA signals at 640 by 480 pixels and popular SVGA displays of 800 by 600 and 1024 by 768, all at 60Hz frame rate.

## How it works

As already mentioned, the micro is the basic building block for the complete design, only a handful of additional parts being required to complete the implementation. All timings are created in software by the microprocessor using an external 40MHz-oscillator clock IC.

Although a crystal and capacitors can be used for the oscillator circuit I have found crystals over 30MHz difficult to obtain. A voltage regulator and a few LEDs are the only additional semiconductors.

My PCB has two buttons, a power connector, a 15-way video connector and a few resistors and capacitors to complete the component count.

The LEDs indicate the display resolution setting, as it is not obvious from

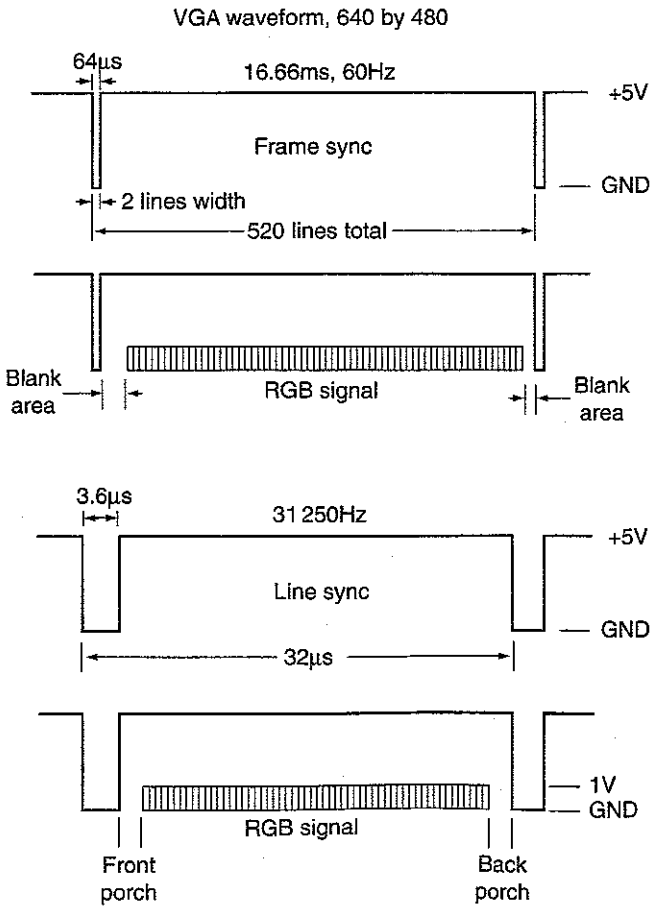



*Fig. 1. Timings for a VGA signal's frame and line sync pulses.*

*Table 1. Forms of the signals needed to drive an SVGA monitor.*

| | |
|---|---|
| Vertical sync | 5V TTL, negative going |
| Horizontal sync | 5V TTL, negative going |
| Red | 1V analogue signal, positive going |
| Green | 1V analogue signal, positive going |
| Blue | 1V analogue signal, positive going |

the monitor display which resolution is being used. Also, if the monitor is not capable of displaying SVGA you will possibly be looking at a black screen.

Although the design can run on batteries and tags are provided on my PCB, I use a ready available 9V mains adapter. After all, you need the mains to power the monitor anyway and batteries have a habit of going flat just when you need them.

There are five signal lines used to drive a standard video monitor, as in **Table 1**. There are also ground returns for all these signals at the connector.

An example of the timings for a VGA signal is shown in **Fig 1**. Vertical and horizontal sync signals are driven directly from the microprocessor. The RGB analogue signals are attenuated by a resistor divider network to give a maximum of 1V output into a 75Ω load.

## Software

The whole design revolves around software running on the Scenix microprocessor.

Timings for the functions are critical. Each scan line must be balanced to within a couple of instructions or tearing of the image will result.

Button presses are checked at the beginning of each frame and a software trap waits for the button to be released for the changes to take place.

The first button changes the displayed output and the second button changes the resolution. Each button press changes the parameters of the main loop which are present in a software table.

The complete code is shown in **List 1**. Note that a special Parallax serial programming adapter is required to program the Scenix parts.

On my PCB, the programmer plugs directly onto the four-way connector next to the micro and the crystal oscillator link removed while programming, **Fig 2**.

## Implementing the design

The unit can be housed in a two-piece plastic moulding with built-in mounting posts for the PCB. All parts can be mounted directly on the board and no wiring needs to be involved in the construction.

As the unit is based on a crystal clock, no set-up procedure or calibration is required. I estimate only a two to three hour construction time if all parts are available. This will of course be extended if you decide to produce your own printed circuit board.

You may socket the microprocessor, but it can be reprogrammed on board making removal unnecessary.

The mains adapter can be regulated or unregulated, 9-12V at 300-500mA. Test the unit by connecting a VGA or SVGA monitor to the output and switching on the unit. The generator will default to colour bars in standard VGA mode of 640*480. Pressing the buttons should change patterns and resolutions.

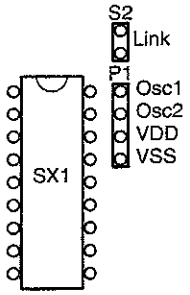
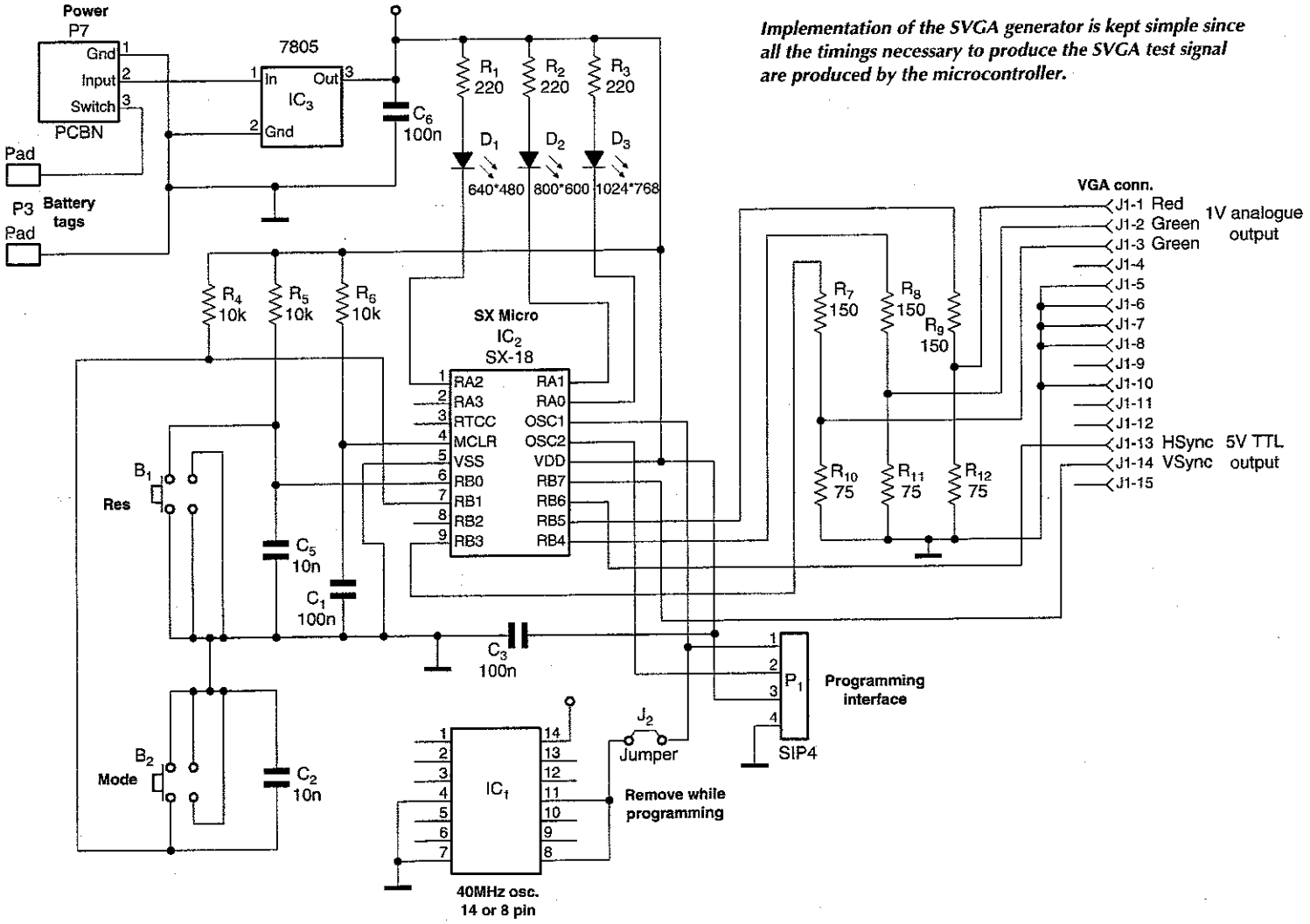



*Fig. 2. On my prototype PCB, the programmer plugs directly on to a four-way connector next to the microcontroller. The crystal oscillator link is removed while programming.*

*Implementation of the SVGA generator is kept simple since all the timings necessary to produce the SVGA test signal are produced by the microcontroller.*



## Components

Resistors 1/8 watt 5% carbon film:

| | |
|---|---|
| $R_{1,2,3}$ | 220 |
| $R_{4,5,6}$ | 10k |
| $R_{7,8,9}$ | 150 |
| $R_{10,11,12}$ | 75 |

Capacitors 50V polyester or ceramic:

| | |
|---|---|
| $C_{1,3,4}$ | 0.1µF |
| $C_{2,5}$ | 0.01µF |

| | |
|---|---|
| $D_{1,2,3}$ | 3mm red leds |
| $B_{1,2}$ | vertical push switch |
| $P_1$ | 4 way 0.1 pitch pins |
| $J_2$ | 2 way 0.1 pitch pins + jumper |

$J_1$ is 15-way min D-type 90°, Maplin part JW85G. Connector $P_7$ is a 3.5mm power connector from Farnell, part 224-959.

$IC_1$ is a 40MHz CMOS or universal oscillator 8-pin or 14-pin, Farnell 704-738 for example

$IC_2$ is the Scenix SX18/AC/DP processor, 18-pin version

$IC_3$ is a 7805 5V positive regulator

Suggested case, Farnell part 250-030. Mains adapter, 9-12V 300mA from CPC.
Approximate build cost £20 plus mains adapter.

## New circuit board service from Quickroute

Electronics CAD specialist Quickroute, in conjunction *Electronics World*, is offering a new printed circuit board service to readers of *Electronics World* .

You can obtain single-sided boards based on Roy's original design direct from Quickroute for a fully inclusive price of £14.69. To order, fax your credit card number and expiry date together with the card-holder's address to 0161 4760505. Alternatively, ring Quickroute on 0161 4760202. If you wish to order by post, write to Quickroute at Regent House, Heaton Lane, Stockport SK4 1BS and send a postal order or cheque payable to Quickroute Systems Ltd.

Overseas readers should contact Quickroute before ordering for details of postage.

Alternatively, you can obtain a disk of the PCB layout in Quickroute form by sending £12.50 via postal order or cheque payable to Reed Business Information to SVGA, Electronics World, Quadrant House, The Quadrant, Sutton, Surrey SM2 5AS.

## Need a programmed controller?

All parts, pre-programmed chips and software on disk can be obtained from the author by sending an SAE marked 'SVGA software' to the Quadrant House address above for details.

## A word of warning

If you try to run a monitor on a resolution that it is not capable of displaying, it is possible to cause damage to its internal circuitry. Therefore, if the display fails to appear on an increased resolution within a few seconds, switch back immediately to a lower resolution.

I used a self-adhesive label on the front panel to make the unit more professional. The artwork was done on a PC and printed out on a colour ink jet printer using glossy self-adhesive paper. ■

***List 1. Complete software listing for the SVGA generator.***

```
;       VGA & SVGA Display
;       colour bar and crosshatch screen displays, by C R Harding
;       Parameters for 40MHz crystal. Finished version 1.01
;port bit 7 frame sync
;port bit 6 hsync
;port bit 5 RED
;port bit 4 GREEN
;port bit 3 BLUE
;port bit 1 Button1
;port bit 0 Button2
                device  pins18,pages2,banks1,oschs,turbo
                device  stackx,optionx,protect
                id      'CRH SVGA'
                reset   start
                org 8
count   ds      1
port    ds      1
temp    ds      1
temp1   ds      1
temp2   ds      1
temp3   ds      1
pattern ds      1
flag    ds      1       ;10h
res     ds      1
colour  ds      1
T1      ds      1
T2      ds      1
T3      ds      1
T4      ds      1
T5      ds      1
T6      ds      1       ;18h
T7      ds      1
T8      ds      1
T9      ds      1
T10     ds      1
T11     ds      1
T12     ds      1
T13     ds      1
T14     ds      1
                org 100h
start   ;setup ports
                mov     !rb,#%00000011          ;port B 6 out 2 in
                mov     !ra,#%00000000          ;port A all out
                mov     port,#255
                mov     pattern,#0              ;colour bars
                mov     res,#0                          ;vga
mainloop
                mov     flag,#0                 ;reset interupt flag
                cje     res,#0,vga      ;vga
                cje     res,#1,svga1    ;svga1
                cje     res,#2,svga2    ;svga2
vga     ;640*480 31250Hz
                mov     T1,#5           ;Front porch delay
                mov     T2,#26          ;Line sync width 3.6uS
                mov     T3,#30          ;Frame blank lines start
                mov     T4,#12          ;Frame blank lines end
                mov     T5,#32          ;Colour bar width delay 26
                mov     T6,#160         ;Number of lines(*3)
                mov     T7,#2           ;number of lines in fsync
                mov     T8,#97          ;line delay length 28uS
                mov     T9,#30          ;chequer lines
                mov     T10,#8          ;chequer squares
                mov     T11,#59         ;crosshatch lines
                mov     T12,#9          ;chequer delay
                mov     T13,#16
                mov     T14,#10
                mov     ra,#%00000011           ;led 3
                jmp     frameloop
svga1   ;800*600 37880hz
                mov     T1,#4           ;Front porch delay
                mov     T2,#23          ;Line sync width 3.2us
                mov     T3,#24          ;Blank lines start
                mov     T4,#8           ;Blank lines end
```

Continued on page 875

```
                mov     T5,#25      ;Colour bar width delay 20
                mov     T6,#200     ;Lines*3
                mov     T7,#4       ;92.8us
                mov     T8,#79      ;line delay length 23.2us
                mov     T9,#25      ;chequer lines
                mov     T10,#12     ;chequer squares
                mov     T11,#49     ;crosshatch lines
                mov     T12,#4      ;chequer delay
                mov     T13,#5
                mov     T14,#13
                mov     ra,#%00000101       ;led 2
                jmp     frameloop
svga2   ;1024,768
                mov     T1,#7       ;Front porch delay
                mov     T2,#13      ;line sync width 2us
                mov     T3,#24      ;Blank lines start
                mov     T4,#4       ;Blank lines end
                mov     T5,#19      ;Colour bar width delay 15
                mov     T6,#256     ;Lines*3
                mov     T7,#6       ;120us
                mov     T8,#59      ;line delay length 19us
                mov     T9,#24      ;chequer lines
                mov     T10,#16     ;chequer squares
                mov     T11,#47     ;crosshatch lines
                mov     T12,#1      ;chequer delay
                mov     T13,#9
                mov     T14,#7
                mov     ra,#%00000110       ;led 0
frameloop
                mov     temp,rb     ;buttons check
                and     temp,#3
                cje     temp,#3,exit    ;exit if buttons not pressed
                call    button
                cje     flag,#1,mainloop    ;jump to mainloop if set
exit    ;64us wide neg pulse vga 106us svga1
fsync
                and     port,#%01111111 ;bit 7 low vertical sync
                mov     rb,port     ;write value to port
                mov     temp,T7
floop
                call    blackline   ;complete horizontal sync line
                djnz    temp,floop
                or      port,#%10000000     ;bit 7 high
                mov     rb,port
;VGA 520 lines total 640*480 display 31.250kHz 60Hz
                cje     pattern,#0,colourbars   ;go routines
                cje     pattern,#1,crosshatch
                cje     pattern,#2,@cheqboard
                cje     pattern,#3,redscreen
;routines must be complete loops, no changes in line timings.
colourbars
                mov     temp,T3     ;frame blank1
b1              call    blackline
                djnz    temp,b1
                mov     temp2,T6    ;160 vga 200 svga
lpa
                call    bars
                djnz    temp2,lpa
                mov     temp2,T6    ;320 vga 400 svga
lpb
                call    bars
                djnz    temp2,lpb
                mov     temp2,T6    ;480 vga 600 svga
lpc
                call    bars
                djnz    temp2,lpc
                mov     temp,T4     ;frame blank2
b2              call    blackline
                djnz    temp,b2
                jmp     frameloop
crosshatch
                mov     temp,T3     ;frame blank1
b3              call    blackline
                djnz    temp,b3
                mov     temp2,T10   ;8 lines vga
cra
                call    whiteline   ;1
                mov     temp1,T11   ;1+59 vga
crb
                call    hatch
                djnz    temp1,crb
                djnz    temp2,cra
                nop
                call    whiteline   ;end line
                mov     temp,T4     ;frame blank1
b4              call    blackline
                djnz    temp,b4
                nop
                jmp     frameloop
redscreen
                mov     temp,T3     ;frame blank1
r1              call    blackline
                djnz    temp,r1
                mov     temp2,T6    ;160 vga 200 svga
r2
                call    redline
                djnz    temp2,r2
                mov     temp2,T6    ;160 vga 200 svga
r3
                call    redline
                djnz    temp2,r3
                mov     temp2,T6    ;160 vga 200 svga
r4
                call    redline
                djnz    temp2,r4
                mov     temp,T4     ;frame blank2
r6              call    blackline
                djnz    temp,r6
                jmp     frameloop
;subroutines lower half of page
                org 0h
linesync    ;3.8us
                and     port,#%10111111
                mov     rb,port
                mov     count,T2    ;delay loop
                call    dly
                or      port,#%01000000
                mov     rb,port
                call    linedly
                call    linedly
                retp
bars    ;7 bars red to white
                cjne    res,#2,bv3
                nop
                nop
                nop
                call    linesync
                jmp     bv4
bv3             nop
                call    linesync
bv4
                or      port,#%00100000     ;red
                mov     rb,port
                call    bardly
                and     port,#%11011111
                or      port,#%00010000     ;green
                mov     rb,port
                call    bardly
                or      port,#%00110000     ;yellow
                mov     rb,port
                call    bardly
                and     port,#%11001111
                or      port,#%00001000     ;blue
                mov     rb,port
                call    bardly
                or      port,#%00101000     ;magenta
                mov     rb,port
                call    bardly
                and     port,#%11010111
                or      port,#%00011000     ;cyan
                mov     rb,port
                call    bardly
                or      port,#%00111000     ;white
                mov     rb,port
                call    bardly
                and     port,#%11000111
                mov     rb,port
                mov     count,T14   ;trim end
trim
                djnz    count,trim
                ret
bardly
                mov     count,T5    ;width delay
boop
                djnz    count,boop
                retp
dly
loop2           nop                 ;delay set in count
                djnz    count,loop2
                retp
linedly
                mov     count,T1    ;porch delay
loop3
                djnz    count,loop3
```

```
                retp
hatch
                cje     res,#1,hv2
                cje     res,#2,hv3
                call    linesync
                mov     temp,#10    ;vertical bars
hl
                or      port,#%00111000     ;white
                mov     rb,port
                and     rb,#%11000111       ;black
                mov     count,#22           ;delay between bars
hllp
                djnz    count,hllp
                djnz    temp,hl
hloop
                or      port,#%00111000     ;white
                mov     rb,port
                and     rb,#%11000111       ;black
                call    linedly
                and     port,#%11000111     ;black
                ret
hv2
                call    linesync
                mov     temp,#6
h2
                or      port,#%00111000     ;white
                mov     rb,port
                and     rb,#%11000111       ;black
                mov     count,#14           ;delay loop
h2lp
                djnz    count,h2lp
                djnz    temp,h2
                mov     temp,#6
h2a
                or      port,#%00111000     ;white
                mov     rb,port
                and     rb,#%11000111       ;black
                nop
                mov     count,#14           ;delay loop
h2lpa
                djnz    count,h2lpa
                djnz    temp,h2a
                jmp     hloop
hv3
                nop
                call    linesync
                mov     temp,#7
h3
                or      port,#%00111000     ;white
                mov     rb,port
                and     rb,#%11000111       ;black
                nop
                mov     count,#8            ;delay loop
h3lp
                djnz    count,h3lp
                djnz    temp,h3
                mov     temp,#7
h3a
                or      port,#%00111000     ;white
                mov     rb,port
                and     rb,#%11000111       ;black
                mov     count,#8            ;delay loop
h3lpa
                djnz    count,h3lpa
                djnz    temp,h3a
                nop
                jmp     hloop
blackline   mov colour,#%00000000
                jmp     line
whiteline   mov colour,#%00111000
                jmp     line
redline         mov     colour,#%00100000
                nop
                nop
                nop
line
                call    linesync
                nop
                or      port,colour
                mov     rb,port
                mov     count,T8            ;delay loop
                call    dly
                mov     count,T8            ;delay loop
                call    dly
                and     port,#%11000111
                mov     rb,port
                call    linedly
                retp
button
                mov     w,rb        ;buttons check
                and     w,#2
                test    w
                jz      reschange
patchange
                add     pattern,#1          ;next pattern
                cjbe    pattern,#3,bexit
                mov     pattern,#0          ;back to start
                jmp     bexit
reschange
                add     res,#1              ;next resolution
                mov     flag,#1             ;set res change flag
                cjbe    res,#2,bexit
                mov     res,#0              ;back to start
bexit
                mov     temp,rb             ;buttons check
                and     temp,#3
                cjne    temp,#3,bexit       ;wait release
                retp
                org     200h
cheqboard
                mov     temp,T3             ;frame blank1
                call    @blackline
b5              djnz    temp,b5
                mov     temp2,T10           ;vertical squares
cheq
                mov     temp1,T9            ;chequer lines
chb
cheque1
                nop
                nop
                call    @linesync
                mov     temp3,T10
ch1
                or      port,#%00111000     ;white
                mov     rb,port
                mov     count,T12           ;delay loop
                call    @dly
                and     port,#%11000111     ;black
                mov     rb,port
                mov     count,T12           ;delay loop
                call    @dly
                djnz    temp3,ch1
                and     port,#%11000111     ;black
                mov     rb,port
                call    @linedly
                nop
                nop
                mov     count,T13           ;trim end
trim1
                djnz    count,trim1
                djnz    temp1,chb
                mov     temp1,T9
cha
cheque2
                nop
                nop
                call    @linesync
                mov     temp3,T10
                nop
ch2
                and     port,#%11000111     ;black
                mov     rb,port
                mov     count,T12           ;delay loop
                call    @dly
                or      port,#%00111000     ;white
                mov     rb,port
                mov     count,T12           ;delay loop
                call    @dly
                djnz    temp3,ch2
                and     port,#%11000111     ;black
                mov     rb,port
                call    @linedly
                nop
                mov     count,T13           ;trim end
trim2
                djnz    count,trim2
                djnz    temp1,cha
                djnz    temp2,cheq
                mov     temp,T4             ;frame blank2
                call    @blackline
b6              djnz    temp,b6
                jmp     @frameloop
```