

H O M E - b r e w e d Issue One: June 7, 1993
the Network Twenty-One Wireheads Information Zine
"Freebirthing the 90's into deep space"
Chicago, IL
Public Access Line at (312) 784-2852

C O N - t e n t s :

o n e: S E G A t h r e e - D I M E N S I O N S
t w o: P A R T - l i s t i n g
t h r: N O W !! t h e C I R C U I T !
f o u: W H A T a b o u t C O D E ?
f i v: W H A T a b o u t S O F T W A R E ?
s i x: t h e P O W E R g l o v e
s e v: W H A T - i s ?
e i g: P O W E R - g l o v e t o I B M P C
n i n: W H A T a b o u t C O D E ?
t e n: t h e B R A I N b o x
0 1 1: t h e A T A R I s t H A C K
0 1 2: t o > A M I G A
0 1 3: N e X t !
0 1 4: W H A T a b o u t T H E m a c - I N T O S H ?
0 1 5: g l o v e E P I L O U G E
0 1 6: V R - s i t e l i s t i n g s
0 1 7: S O F T - w a r e !
0 1 8: M U L T I V E R S E
0 1 9: G O S S A M E R
0 2 0: F L Y !
X X I: R E N D - t h r e e - E I G H T Y - s i x
0 2 2: O T H E R - p r o g r a m s
0 2 3: d i s - C L A I M

B E - g i n

S E G A - t h r e e - D I M E N S I O N S

First things first, the Sega 3d glasses DO NOT ACCEPT A VIDEO SIGNAL. You do not "feed" them NTSC or VGA, they are "shutter glasses". One goes on as the other goes off, and vice versa. Your system will simulate two different views of the scene, thus, when the glasses perform their "shutter" action, you are actually watching two different images, one out of each eye and one at a time, very quickly. This creates the S T E R E O S C O P I C effect.

--

The sega shutter glasses as explained to me by Ross Leonard:

Sega shutter glasses do not display an image. Instead, each LCD panel (1 per eye) alternates on or off, therefore blocking light that can pass through the respective lens. The computer (or Sega system) alternately displays left and right side images on the screen that are synchronized to the glasses, creating a rough stereoscopic view. When the computer displays a right-side image, the left "shutter" is closed, so you can only see it with the right eye, etc. I'm not sure of the exact display rate, but since there is a slight flicker, I would assume it would be in the 15 to 20

per second range.

Ross Leonard

The gods then said "You will need parts to complete your circuit."

P A R T - l i s t i n g

RSPTN= Radio Shack part number

NM = Part name

QTY = Quantity in package

PCN = Packages needed

\$\$\$ = Subtract this figure from net worth after purchase

(Network 21 not responsible for errors in figures, and is not affiliated with the Radio Shack corporation)

RSPTN	NM	QTY	PCN	\$\$\$
276-1617	2N2222 transistor	15	1	1.98
271-1335	10K 1/4 watt resistor	5	1	.39
271-1339	22K 1/4 watt resistor	5	1	.39
272-131	.01uF capacitor	2	1	.49
272-1026	22uF capacitor,	?	1	.69
unknown	Rectifier diode	2	3 needed	.49-.79
276-175	Breadboard	1	1	7.49

(RCA CD4030) Quad XOR gate (not available at most Chicago area RS)

___ Quad XOR gate will most likley have to be special ordered.

Soldering iron & solder

Cable for appropriate RS-232 port (to be cannibalized)

Stereo jack (to be cannibalized)

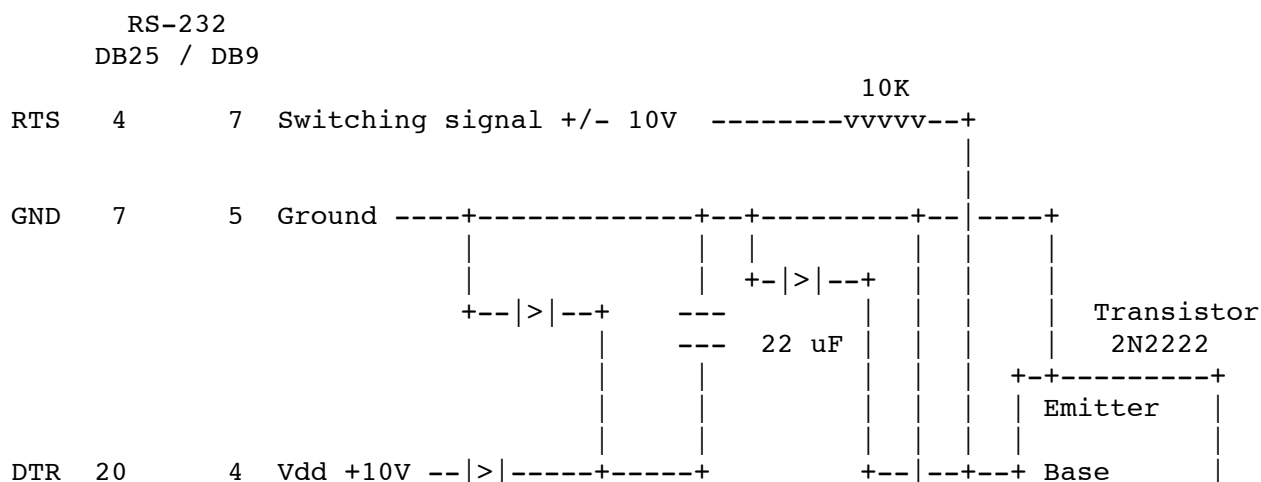
Well, there you have it, teenage america.

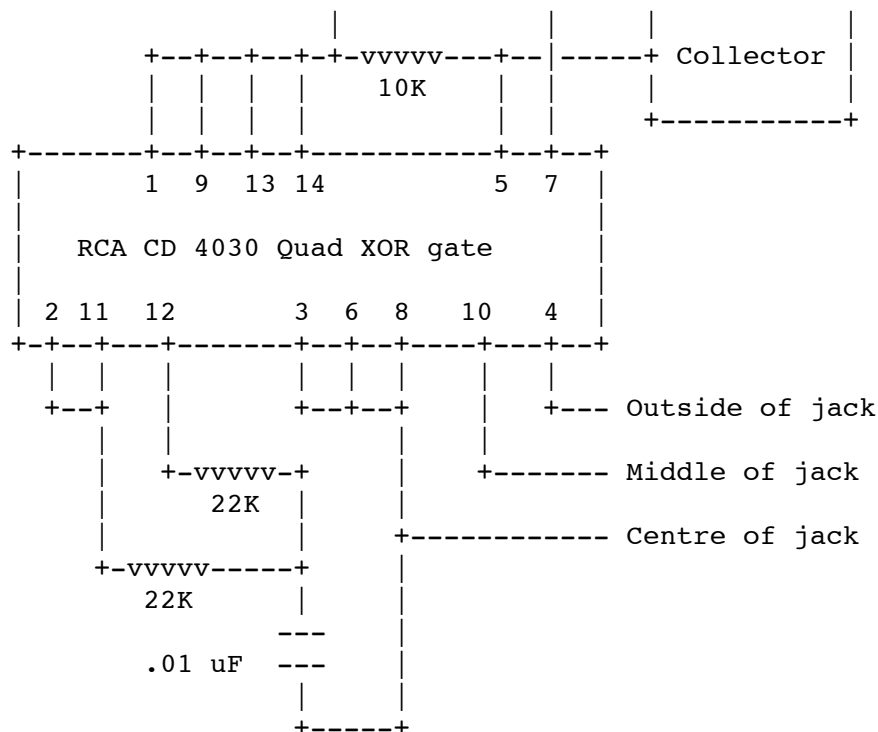
(Part no. lookups courtesy of gsumners@ntwrk21.chi.il.us.)

N O W !! t h e C I R C U I T !

Sender: hlab@milton.u.washington.edu (Human Int. Technology Lab)

Circuit to connect SEGA 3D glasses to RS-232 port.





Diode: --|>|--

Resistor: --vvvvv--

Capacitor:

|

W H A T a b o u t C O D E ?

The following code was written by F van der Hulst at a space-time fix of Feb.2, Aut 91.

It is source for viewing three-dimensional images with the Sega glasses and your constructed circuit.
Relevant info within comment text.

/*

VGA 320 * 400 * 256 * 2 frames routines.

Written by: F van der Hulst, 20/2/91

These routines display pixels in 320*400 mode by modifying the VGA registers, as outlined in Programmer's Journal V7.1 (Jan/Feb '89) article, pages 18-30, by Michael Abrash.

The advantage of 320 * 400, is that it gives two separate video pages, which can be displayed on the screen independently. These can contain two views of a scene, taken from slightly different viewpoints. These are displayed alternately on the screen, in sync with a pair of "chopper glasses", to give a 3D effect.

*/

```

#include <conio.h>

typedef unsigned char DacPalette[256][3];

/* Setvgapalette sets the entire 256 color palette */
/* PalBuf contains RGB values for all 256 colors */
/* R,G,B values range from 0 to 63 */
/* Taken from SVGA256.H, by Jordan Hargraphix Software */

void setvgapalette(DacPalette *PalBuf)
{
    struct REGPACK reg;

    reg.r_ax = 0x1012;
    reg.r_bx = 0;
    reg.r_cx = 256;
    reg.r_es = FP_SEG(PalBuf);
    reg.r_dx = FP_OFF(PalBuf);
    intr(0x10,&reg);
}

unsigned int act_page = 0; /* Current page being written to */

#define VGA_SEGMENT      0xa000
#define SC_INDEX         0x3c4
#define GC_INDEX         0x3ce
#define CRTC_INDEX       0x3d4
#define DISPIO           0x3DA

#define MAP_MASK         2
#define MEMORY_MODE      4
#define GRAPHICS_MODE    5
#define MISCELLANEOUS    6
#define VRT_bit          8
#define MAX_SCAN_LINE    9
#define START_ADDRESS_HIGH 0x0c
#define UNDERLINE        0x14
#define MODE_CONTROL     0x17

void writepixel(int x, int y, unsigned char colour)
{
    long addr;

    addr = ((x >> 2) + 320/4 * y + act_page);
    addr = ((addr & 0xffff0000L) << 4) + (addr & 0xffffL) + ((long) VGA_SEGMENT << 16);
    outport(SC_INDEX, (0x100 << (x & 3)) | MAP_MASK);
    *(char far*)addr = colour;
}

void set320x400mode(void)
{
    struct REGPACK regs;
    unsigned char x;

    regs.r_ax = 0x13; /* Set 320*200*256 graphics mode via BIOS */
    intr(0x10, &regs);
}

```

```

/* Change CPU addressing of video memory to linear (not odd/even, chain, or
now chain 4), to allow access to all 256K of display memory. Each byte will
1 control one pixel, with 4 adjacent pixels at any given address, one pixel
per plane. */

outportb(SC_INDEX, MEMORY_MODE);
x = inportb(SC_INDEX+1);
x &= 0xf7;
/* Turn off chain 4 */
x |= 4;
/* Turn off odd/even */
outportb(SC_INDEX+1, x);
outportb(GC_INDEX, GRAPHICS_MODE);
x = inportb(GC_INDEX+1);
x &= 0xef;
/* Turn off odd/even */
outportb(GC_INDEX+1, x);
outportb(GC_INDEX, MISCELLANEOUS);
x = inportb(GC_INDEX+1);
x &= 0xfd;
/* Turn off chain */
outportb(GC_INDEX+1, x);

/* Now clear the whole screen, since the mode 13h set only clears 64K. Do this
before switching CRTC out of mode 13h, so that we don't see garbage on the
screen. */

outport(SC_INDEX, 0x0f00 | MAP_MASK); /* Write to 4 planes at once */
setmem(MK_FP(VGA_SEGMENT, 0), 0xffff, 0);

/* Change mode to 320*400 by not scanning each line twice. */
outportb(CRTC_INDEX, MAX_SCAN_LINE);
x = inportb(CRTC_INDEX+1);
x &= 0xe0;
/* Set maximum scan line to 0 */
outportb(CRTC_INDEX+1, x);

/* Change CRTC scanning from doubleword to byte mode, allowing the CRTC to
scan more than 64K */
outportb(CRTC_INDEX, UNDERLINE);
x = inportb(CRTC_INDEX+1);
x &= 0xbf; /* Turn off doubleword */

outportb(CRTC_INDEX+1, x);
outportb(CRTC_INDEX, MODE_CONTROL);
x = inportb(CRTC_INDEX+1);
x |= 0x40; /* Turn on the byte mode
bit, so memory is linear */
outportb(CRTC_INDEX+1, x);
}

void end320x400mode(void)
{
struct REGPACK regs;

regs.r_ax = 3; /* Return to text mode */
intr(0x10, &regs);

```

```

}

/* Set visible page */

void setvispage(int page)
{
    outport(CRTC_INDEX, (page << 15) | START_ADDRESS_HIGH);
}

/* Set active page (page being written to */

void setactpage(int page)
{
    act_page = page ? 0x8000 : 0;
}

void WaitForVerticalRetrace(void)
{
    static char chopper = 1;

    while (inportb(DISPIO) & VRT_bit) /* wait */ ;
    while ((inportb(DISPIO) & VRT_bit) == 0) /* wait */ ;
    if ((chopper++ & 1) == 0) outportb(0x3fc, 1);
    else
        outportb(0x3fc, 3);
}

void main(int argc, char *argv[])
{
    set320x400mode();

    /* Now fill the rgb_palette structure in memory with colour info */

    setvgapalette(&rgb_palette);

    setactpage(0);
    /* Now call writepixel to put stuff on page 0 */
    setactpage(1);
    /* Now call writepixel to put stuff on page 1 */

    while (!kbhit()) {
        WaitForVerticalRetrace();
        setvispage(0);
        WaitForVerticalRetrace();
        setvispage(1);
    }
    getch();
    end320x400mode();
}

--

W H A T   a b o u t   S O F T W A R E ?

- rend 386

```

Quoted from documentation:

"This version now support stereoscopic viewing; the assumption is that you have the Sega 3D glasses

```

-x    ensable stereo (use if you don't have sega glasses)
-m    use mirror stereo
-r    reverse eyes (left-for-right); useful if your wiring is wrong
-l    use COM1 for Sega glasses
-2    use COM2 for Sega glasses"

```

--
 Rend 386 by Dave Stampe and Bernie Roehl is available via anonymous ftp
 from sunee.uwaterloo.ca in the directory "/pub/rend386"
 --

"There is also a mailing list, rend386@sunee.uwaterloo.ca (to be added
 to the list, send mail to rend386-request@sunee.uwaterloo.ca). Traffic
 should be reasonably low; if it gets too high, we're willing to go to a digest
 format."

Bernie Roehl

—
 The demonstration and libraries are also available on Network 21.
 It is worth checking out.
 It REQUIRES at least a 386 or 486, and standard VGA. It -will not- run on
 an 80286.
 Rend 386 also supports the Mattel Powerglove as a 3d input / object
 manipulation device (it is neat, and -fast-!)
 Very "C like". Very powerful.
 Two nodes up.
 —

t h e P O W E R g l o v e

Included:

Connections to the IBM PC, Atari ST, NeXt, Amiga, info on the Power Glove
 mailing list, source code for the IBM PC, Amiga, ST, and sources for Mac
 Power-Glove / Sega information.

W H A T - i s ?

The Mattel Powerglove was originally intended as a device for making
 your video-game life more enjoyable. Manufactured for the Nintendo
 Entertainment system, it originally carried a price tag of \$99.00+.
 Now it has been adopted by the street as a 3d input device for a
 wide-range of personal and super-computers. I personally think that
 you get better control with the mouse, but there are disadvantages to
 the mouse: two dimensions, you can't wear it, etc.
 Code is now widley available for use with the power-glove, as well as
 third party software such as the forementioned rend 386.

The following are instructon texts gathered from various places on "hooking
 in" your perhaps lawnmower-man inspired input device:

P O W E R g l o v e - t o I B M P C

Be sure and check out the BYTE magazine article in the subject line,
 (issue: July, 1990;page 288). This article provided widespread inspiration
 for the PG's current "home brewed" vr use. Illustration included.