1990

Lasers & Computers: Graphics with a Flash

J. Daniel Ashton

Lasers & Computers: Graphics with a Flash

All my life I've been enthralled with flashy graphical displays. While most people ignore the commercials and the opening and closing graphics on television, I usually like them better than the main programming. The Stone Mountain Laser Show struck me as another of those wonderful graphics. While I enjoy the program for its artistic content and inventiveness, my mind is dumbstruck by the reality of pure laser light being controlled by computer in such a beautiful manner.

Since the year I first saw that show I have wanted to be able to control those flashing beams, and through the intervening years I have dreamed of perhaps taking such a show on the road. As a senior at Collegedale Academy I was privileged to visit the workshop at Stone Mountain, and get a closer look at the equipment and software operating the show. And I kept on daydreaming about actually doing that sort of thing myself.

Somehow, in spite of fiscal reality, that dream never quite died. And as I came through college the dream transferred itself into plans for my Senior Honors Project. Everyone told me how unrealistic I was being, wanting to combine personal computers and synthesizers and slides with a flashing laser beam. And they were right. But this year has seen a partial fulfillment of that dream. This project may be all I ever do with this dream, and the resemblance between the project

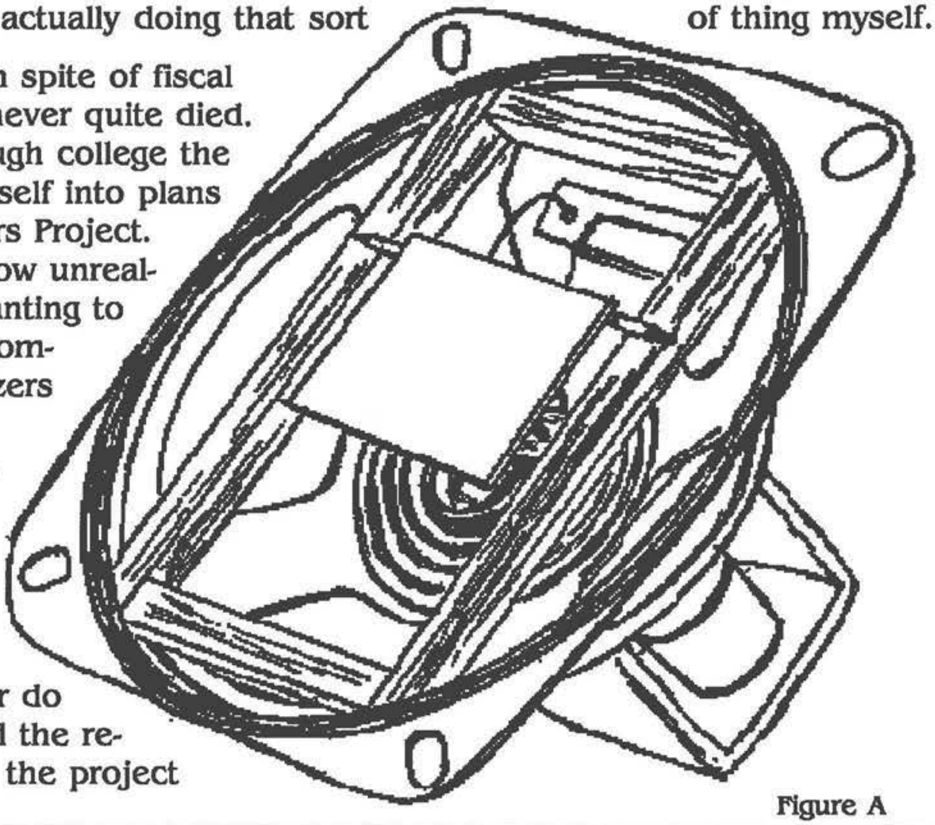

Figure A

and my dream is certainly very slight. But the experience of actually doing it has been an unexpected thrill.

The project, up to this point, has been mostly a graphics hardware construction project. The resulting hardware consists of two (mangled) speakers, two front-surface mirrors, a small amount of balsa, wire and plastic, about twenty resistors, fourteen transistors, two 8212 IC chips, and a bunch of wire. The speakers, wire, plastic, balsa and mirrors are assembled as shown in **figure A**, and as a result, current applied to the speaker leads causes the mirror to vibrate on one axis, much as it would have caused the cone (now cut out) to vibrate and produce sound.

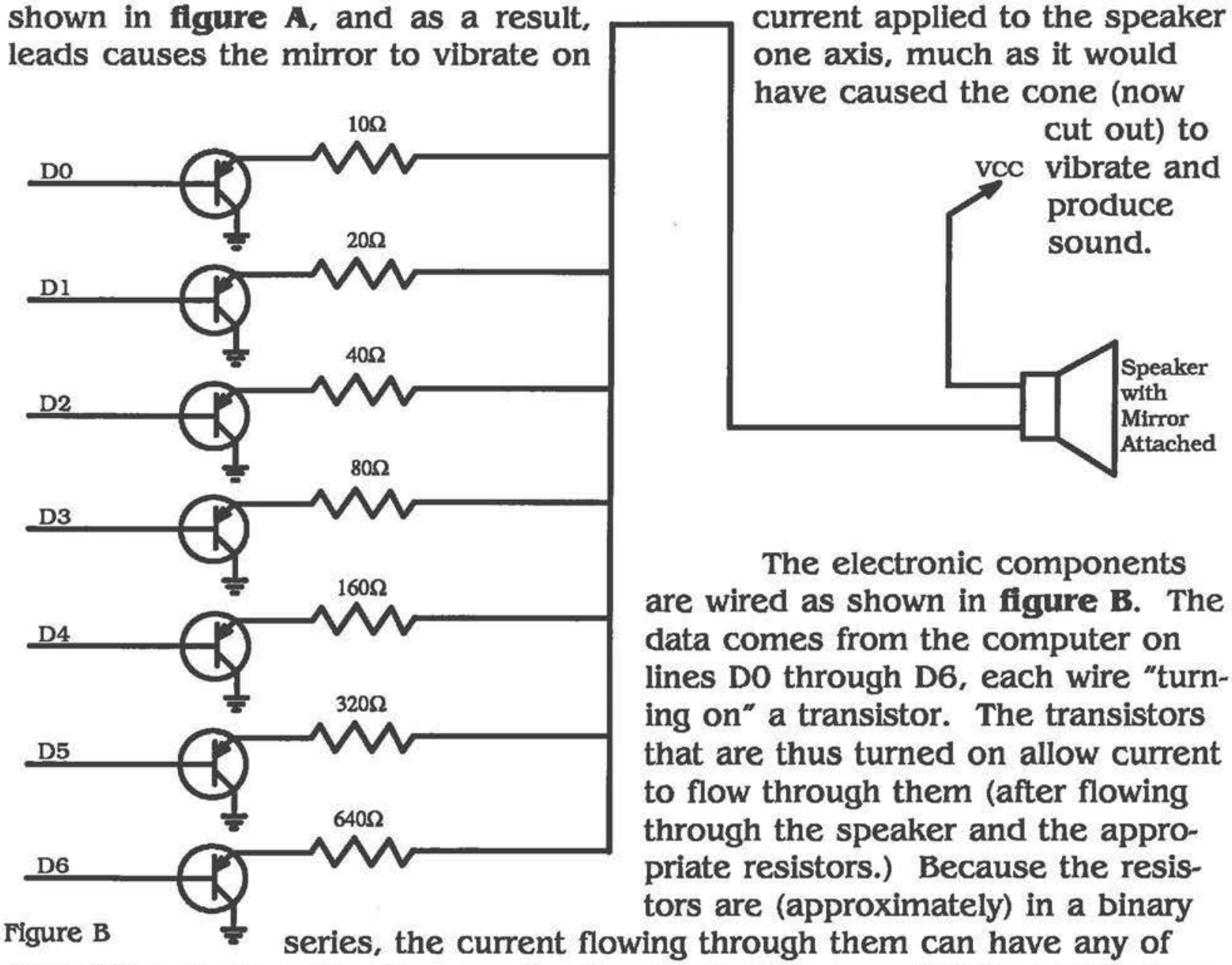

Figure B

The electronic components are wired as shown in **figure B**. The data comes from the computer on lines D0 through D6, each wire "turning on" a transistor. The transistors that are thus turned on allow current to flow through them (after flowing through the speaker and the appropriate resistors.) Because the resistors are (approximately) in a binary series, the current flowing through them can have any of 128 different strengths, thus moving the speaker to one of 128 possible locations.

The two speakers are mounted as shown in **figure C**. The lower speaker, because of its angle and the angle of rotation of its mirror, controls the horizontal motion of the laser beam. The top speaker controls the vertical motion.

Mathematics has, in many ways, been the basis for all electronic circuitry, and in this way it has a very important part in this project. However, that part of mathematics is the domain of other fields of study, and there is not time to detail it here.

The binary series is one mathematical reality that makes computers in general a feasibility, and plays an important part in this project. Numbers (as we think of them) can be encoded in a series of on/off switches. If we write 0 for a switch that is off and 1 for a switch that is on, then seven switches can hold for us 128 different values, from 0000000 and 0000001 to 1111110 and 1111111. This binary series can be described by any set of objects that can have only two states. A set of wires that can either have or not have current flowing through them can represent this binary series, and the data pins on the parallel port of a personal computer operate on this basis.

As mentioned above, I have connected these data pins to the seven transistors in my circuit, and the current flowing or not flowing through these pins switches the transistors either on or off. When the transistor is on it allows current to flow through it from its resistor. When the transistor is off the current is blocked.

The resistors also form a binary series, with the lowest resistance being 10 ohms, the next 20 ohms, the next 40 ohms, and so forth up to 640 ohms. The current flowing through these resistors is thus regulated into one of 128 strengths. This current "drives" the speaker, and the speaker is forced to a certain position by the strength of the current. Theoretically, this means that the speaker will only come to rest at one of 128 discrete locations.

The software portion of this project is still very rudimentary, mainly because the hardware has presented so many problems that there has not been time to fully develope appropriate software. The few programs that do use this hardware are all based on a very simple algorithm:

```
begin
    initialize the data structures;
    calculate all the points in the figure;
    repeat
        display the figure
    until a key is pressed;
end.
```

The main data structure, which holds all the points, is a dynamically allocated array of records. The record is made up of two bytes, one each for the X and Y coordinate values of each point. As originally planned these would be the endpoints of the lines in the figure (this **is** supposed to be vector graphics,) but at present the lines calculated are

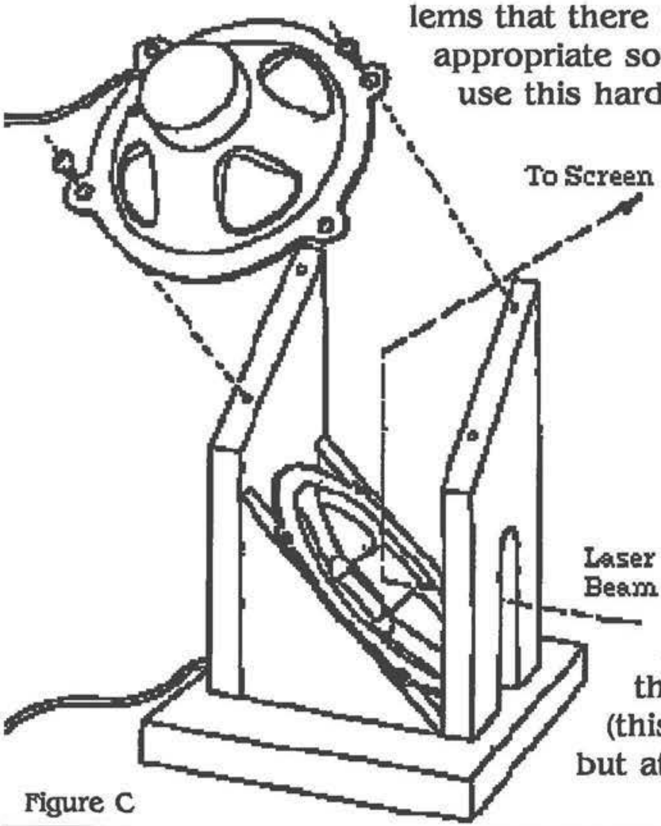

Figure C

made up of every point on the line, for reasons discussed below.

The main code allocates memory for the array, and sets the number of points to zero. The next step in the algorithm is performed by three different procedures. The first simply places a point as a beginning point for a figure component (square, triangle, whatever.) The second calculates all the points between the last point and the one passed to it, and adds them each to the array. The third routine specializes in drawing circular arcs, adding a point for each thirty minutes of arc. The final part of the algorithm is a procedure which runs a **for** loop to display all the points, cycling until a key is pressed.

Problems have abounded in developing this system. Among the first was the speaker/mirror construction. Some inspiration for the design was taken from an article in **Scientific American** by Jearl Walker. Finding small silvered mirrors seemed to be a problem, so I went to the Biology department, and one of the teachers gave me about two dozen slide covers. These are one-inch square pieces of glass less than one sixty-fourth of an inch thick. The speakers were from a used car-stereo. Walker's article mentioned gluing the mirrors to the center of the speakers and to an edge mount, and so I did. I glued a toothpick into place in each speaker, and glued a slide cover to it and to the center of the speaker, using a silicon caulk as glue for "maximum" flexibility. This arrangement presented some interesting patterns, but I felt that the mirrors were not moving far enough, and, because there was a large amount of resonance in the system, I suspected that they were bending, waving perhaps, rather than rotating as needed. By resonance I mean that the beam overshot the point (usually on both axis) and came back, swinging closer to the point each time until it finally sat still. This all happened much to fast for the eye to follow, but was visible as a solid pattern which seldom even resembled what I was trying to produce. These patterns looked beautiful, but not controlled!

I pulled the toothpicks out, and with them the mirrors. Sure enough, the mirrors were quite stiff on the toothpicks. They probably had not rotated very much at all. I replaced the toothpicks with round dowels, hoping that the decreased area of tangency would allow more rotation and less bending. I also glued the center, rather than the edge, of the mirror to the center of the speaker (elevated by a bit of styrofoam), which effectively doubled the distance the mirror moved. This system still showed a lot of resonance, and I felt that some of it was due to the stiffness of the mirror on the dowel. Indeed, when I pulled out the dowel the mirror stayed quite stiff on it. The mirror had probably been bending again rather than rotating.

At a hobby store in Chattanooga I found some lightweight nylon hinges designed to hinge ailerons on model airplane wings. With some balsa I constructed a one-inch square frame to mount the hinges in and the mirror on. The other side of the hinge I glued to the edge of the speaker. I connected the center of the speaker to the frame with a ball-and-socket joint from the hobby store. This system showed a distressing amount of play, horizontal on the vertical mirror, vertical on the horizontal mirror. As far as I can tell the nylon hinges had a certain amount of play built in, and the nylon itself bent readily.

As a final attempt at solidity I worked up the system shown in **figure A**. It uses some stiff wire with a close-fitting plastic sleeve for the hinge, and a beam glued across the center of the speaker holds another hinge for the middle of the mirror, making both hinges very solid two-point connections. About this time my mother suggested checking with a camera repair shop for small mirrors, and I was able to obtain two fairly lightweight front-surface mirrors. These were both stiff enough to resist bending, and they reflected the beam much more strongly than the unsilvered slide covers.

This arrangement did correct the cross-axial play problem, and I think it has helped to alleviate the resonance problem. However, there is still quite a bit of resonance in the system, especially when moving across large distances. This, I suspect, is partly due to the increased mass (and therefore inertia) of the mirrors. However, I have not come up with a good way to damp it.

Another problem that has showed up early in the testing was one of non-linearity. Specifically, the points near the point (0, 0) were spaced much farther apart than those points near (127, 127). My best guess was that in calculating the values for the series of resistors I needed to add in the resistance of the speaker, which is near 10 ohms (the same as the first data bit.) If this is the case, then the series of resistances, is 20, 30, 50, 90, 170, 330, 650, rather than 10, 20, 40, 80, 160, 320, 640. The difference becomes trivial towards the right end of that series. However, on the left end it makes a very big difference. This could explain some of the difference in distance between points at opposite ends of the display area.

The obvious solution to this problem was to either increase the resistances of the series, or reduce the resistance of the speaker. The second was easier to do, and so I soldered three 10 ohm shunt resistors across the leads of each speaker, essentially reducing its resistance to about 2.5 ohms. This seems to have alleviated the problem, but it also reduces the movement of the mirror. This reduced movement, however, has been hardly noticeable, which suggests that the current passing through the speaker was too strong for it. This strong current could also have caused the non-linearity problems observed. If the first step in the series made the speaker move one-half its maximum movement. . . .

These conclusions were supported by using an oscilloscope. While I was working on the final mirror mounting Bradley Hyde suggested that I temporarily replace the mirrors and laser with an oscilloscope, which could read the same values and produce a similar display. In place of the speakers (which had a resistance of ten ohms) he suggested wiring a ten ohm resistor. The oscilloscope was connected between the speaker and the resistor series. The oscilloscope displayed the same type of non-linearity I had observed with the speakers. When I replaced the ten ohm resister with a one ohm resistor the display became much more linear, but also much smaller. In addition, it took on the shape of a parallelogram, rather than a square, with the long diagonal being the one from (0, 0) to (127, 127). This I do not understand.

The oscilloscope helped pinpoint several problems with the hardware. One of these was the problem with untrue resistors. At the beginning of the project I was unable to acquire resistors that fell perfectly along the binary series. Instead I had to choose values close to those I needed, occasionally using a pair of resistors to get nearer the desired value. As it is now the resistors are still not on value. The worst, I think, is the pair that should be 320 ohms, but instead read about 300 ohms. These differences are not large enough to destroy the system, but they does cause holes in a straight line. Twice in the line the points jump back about three places, and I suspect that this is the reason.

The oscilloscope also suggested that the X and Y values were not being written simultaneously. (This could be seen in the fact that lines between points did not seem to attach the points, but moved, somewhat at random, across the display.) In the initial version of the program I used DOS to write to the two parallel printer ports to which the X and Y controlling circuits are attached. This proved to be a very slow way to control the mirror movement, and DOS returned an error every time I wrote to LPT1. (I still don't know why.) This error was solvable by disabling I/O checking and calling IOResult after every write to LPT1. Turbo Pascal provides direct access to the CPU ports through its PORT array, and this turned out to be a better solution, first because it bypasses DOS and thereby gains a few orders of magnitude in speed, and second because it only writes to the Data lines, and never checks for an I/O error.

Even with this improvement the lines on the oscilloscope did not connect the points, although my mind would like to believe that they travelled less randomly. My brother Ted suggested that I needed to latch the outputs so that they did occur simultaneously. I had thought of this before, but had not known of a way to do it. Ted recommended the use of two 8212 IC chips, and Mr. Hyde provided these chips, along with breadboard and the necessary knowledge. After wiring these chips into the circuit between the computer and the transistors I can demonstrate that the X and Y values change at approximately the same time, when I "strobe" the chips. However, the lines on the oscilloscope still do not connect the points. I am guessing that while the 8212's are strobed, their output becomes undefined, causing the wavy lines on the oscilloscope. I haven't been able to try this new system with the laser/mirror combination as of this writing.

Using the oscilloscope has led me to temporarily discard the principles of true vector graphics and use the display as though it needs to have every point drawn. As I mentioned above, the oscilloscope shows no direct connection between points, but rather it shows a set of points with trails from each wandering off into who knows where. To combat this I turned down the intensity until the (non-) connecting lines disappeared, and only the dot were visible. I rewrote the program to draw all the points in the picture, and the display (though distorted) is recognizable. I have not had a chance to try these things on the mirror/laser combination, but I suspect that this will greatly reduce the resonance problem (because most of the steps are very small) and that the laser display will look somewhat less jagged than the oscilloscope.

As can easily be seen, this system still has a long way to go. I have already mentioned the problems with the speakers resonating and the resistors being slightly untrue. In addition, the speakers are rather slow, and the high speed necessary to keep a 30 frame per second display rate may just be impossible. Also, the speakers are only being driven in one direction, and they are being allowed to move in the other direction powered only by their elasticity. This means that any given point may appear in two (or more) different spots on the screen, depending on the point from which you are approaching it. In addition, with the current hardware you can only draw continuous line drawings, since there is no way to block or disable the beam while moving across the screen.

Software problems will arise also. For instance, it presently takes all the CPU's attention to drive the display. In an actual application the program would need time to calculate and/or load more pictures while the current one was being displayed. There should be a smooth user interface, allowing the user to draw a picture by connecting points onscreen, and displaying the created figure as the user draws it. And the possibilities go on and on: animation, music synchronization, *et cetera*.

Most of these problems have solutions, although I am still wondering about some of them. Specifically, the speaker-resonance problem remains a conundrum, as does the problem of speed. One option may be to redesign the system to use stepping motors, rather than voice-coils (the speakers) to move the mirrors. The continuous line drawing problem may be solved by a speaker with a blocking material which is driven either forward or backward, either blocking the beam or allowing it to pass.

The software problems appear to be less difficult to surmount. Good routines exist for almost any graphic application desired, and adaptation to vector graphics (if the hardware can actually do good vectors) usually simplifies the routine. MIDI boxes are available which will help to facilitate synchronization with music. And if the hardware is fast enough (and vectors, with their reduced number of point, are available) a device driver could rather easily be implemented to interrupt the processor thirty times a second just long enough to refresh the display.

All in all, I feel good about what I have accomplished with this project. It is (as far as I can see) not practically usable in its present form. But it has been fun. And what it does now is beautiful, if not controlled. The things I have learned about transistors, resistors, voltage, current, IC chips, *et cetera*, (**ALL** credit due to the patience of Bradley Hyde) will probably come in useful at some unforeseen point in the future. And the books I have browsed have proven interesting.

References

Ideas for the hardware design came from two articles by Jearl Walker in the Amateur Scientist section of **Scientific American**, one in August of 1980, and the other in January of 1981.

The graphic routines I am presently using are adapted from our textbook, **Computer Graphics: the Principles behind the Art and Science** by Cornel Pokorny and Curtis Gerald. I have found this book to be a storehouse of ideas to delight my mind, and I am eagerly looking forward to having evenings free to work through this book, trying everything out and reveling in the results.

The Turbo Pascal and Turbo Assembler manuals have been invaluable in working out the technical details of the software. I especially needed them to learn about the ports, and about the instructions used to write to them.

Ideas for software improvement along the line of user interface came from **Human Interface Guidelines: The Apple Desktop Interface** and from **Building User Interfaces by Direct Manipulation**, a research report by Luca Cardelli, produced by DEC's Systems Research Center.

Ideas for the feasibility of a device driver came The Waite Group's **MS-DOS Papers**, a collection of reports on various unknown facets of MS-DOS and IBM-PC clones. This book has a wealth of information that is difficult to obtain anywhere else, including many undocumented routines and tricks. If there were any one book I would steal from the Daniel's Hall computer library upon graduation, it would be this one. As it is, I'll probably have to be content with purchasing one.