THE UNIVERSAL

Use this handy BASIC program to transform abstract equations into tangible graphs!

BY JAMES E. TARCHINSKI

Over the years "A picture is worth a thousand words" has become just another trite cliche. But in the world of electronics, it certainly is the truth.

As an example of how valuable pictures can be in electronics, consider for a moment the schematic diagram. A schematic is nothing more than a pictoral representation of an electronic circuit. Without such diagrams, we would be forced to describe even the simplest of networks by using pages and pages of text: "A ¼-watt, 330-ohm resistor is connected between the output buffer of the 555 timer and the positive 5-volt terminal of the main power supply..."

The Graph. Schematics aren't the only pictures that aid the electronic hobbyist; there are also graphs. Graphs are excellent for conveying the relationship between two or more variables, such as how a voltage changes with respect to time in an AC network. Essentially, graphs transform very abstract mathematical equations into a visual pattern that our minds can easily process and comprehend.

While graphs are generally very easy to interpret and understand, they are not always so easy to create. Starting with a blank sheet of paper, you must first draw the two axis and divide each of them into an appropriate scale. Next you plot anywhere from five to fifty, or more, points, depending on exactly what it is that you are trying to graph. Lastly, you connect the points with a smooth curve and hope that the end result is worth the time it took to draw. Sometimes it's worth the effort and you are pleased with the results. Other times, however, you end up starting the process over and thinking that there has to be a better way. Well, now there is a better way: *Grapher.Bas*!

LISTING 1—GRAPHER.BAS

```
1000 'GRAPHER.BAS PROGRAM FOR THE PC
1010 '
1020 CLEAR : SCREEN 0,0,0,0 : COLOR 10,0,0
1030 WIDTH 80 : CLS : KEY OFF
1040 PI = 3.14159265
1050 '
1060 PRINT "*****************************************************************"
1070 PRINT "*                                                               *"
1080 PRINT "*                          GRAPHER.BAS                          *"
1090 PRINT "*                                                               *"
1100 PRINT "*               (c) 1988 by James E. Tarchinski                 *"
1110 PRINT "*                                                               *"
1120 PRINT "*****************************************************************"
1130 COLOR 11
1140 PRINT
1150 PRINT "    This program is a plotting utility that allows users to"
1160 PRINT "plot mathematical functions.  If you wish to print out these"
1170 PRINT "graphs, you may do so by using the 'SHIFT-PrtSC' function"
1180 PRINT "of the computer."
1190 PRINT
1200 PRINT "    Before this program is run, however, you must modify the"
1210 PRINT "print values in Lines 7000 - 7160.  You must also define the"
1220 PRINT "two functions that you wish to plot, which may be entered as"
1230 PRINT "subroutines starting at Lines 8000 and 9000."
1240 PRINT "
1250 PRINT "    If you have not yet made these modifications, press the"
1260 PRINT "'E' key to exit the program...."
1270 PRINT "
1280 LOCATE 23,1 : COLOR 7 : PRINT "Press any key ('E' to break)...";
1290 '
1300 IN$=INKEY$:IF IN$<>"" THEN GOTO 1300
1310 IN$=INKEY$:IF IN$= "" THEN GOTO 1310
1320 CLS : IF IN$="E" OR IN$="e" THEN END
1330 '
1340 '
1350 '********** INITIALIZE VARIABLES & SCREEN ************
1360 '
1370 SCREEN 2 : CLS  'enter graphics mode
1380 LINE (76,12)-(76,172)
1390 LINE -(636,172)
1400 '
1410 FOR I=76 TO 636 STEP 56 : LINE (I,170)-(I,174) : NEXT I
1420 FOR I=12 TO 172 STEP 16 : LINE (74,I)-(78,I) : NEXT I
1430 '
1440 GOSUB 7000    'get graph values
1450 '
1460 IF LEN(T1$)>60 THEN T1$=LEFT$(T1$,60)   'limit to 60 characters
1470 IF LEN(T2$)>60 THEN T2$=LEFT$(T2$,60)
1480 IF LEN(X$)>60 THEN X$=LEFT$(X$,60)
1490 C=44-INT(LEN(T1$)/2) : LOCATE 1,C : PRINT T1$;
1500 C=44-INT(LEN(T2$)/2) : LOCATE 2,C : PRINT T2$;
1510 C=44-INT(LEN(X$)/2) : LOCATE 24,C : PRINT X$;
1520 '
1530 L=LEN(Y$) : IF L>18 THEN L=18
1540 FOR I=1 TO L
1550   LOCATE I+2,1 : PRINT MID$(Y$,I,1);   'print y-axis lable
1560 NEXT I
1570 '
1580 DY=(YMAX-YMIN)/10
1590 FOR I=0 TO 10       'lable y-axis loop
1600   J=YMIN+I*DY
1610   LOCATE (22-2*I),2 : PRINT USING "+####.##";J
```

Grapher is BASIC-language program for PC's that plots one or two mathematical functions on a high resolution screen. There are no scales to calculate, no points to plot by hand, and no curves to draw. All you need to do to get high-quality graphs is to modify several constants in the program, enter the mathematical equations to be plotted, and then run the program—Grapher will do the rest.

Using Grapher. To use Grapher, load BASIC into your computer, enter Listing 1, and immediately save the program to disk to protect yourself in case of a system crash. After loading the program and typing in RUN and RETURN, you should see a page of text appear on the screen. In it are instructions on how to use the program. For testing purposes, disregard the warning about making modifications and press the space bar to continue.

If you have entered the program correctly, you should see the text page replaced with a graph carrying the very technical sounding title: "**THIS IS THE FIRST TITLE LINE OF THE GRAPH,**" followed by another title line. The title lines are a good example of the program's frills. When using the program, you will replace those lines with the title of your particular graph. You

LISTING 1 (continued)

```
1620 NEXT I
1630 '
1640 J$=STR$(XMAX) : LOCATE 23,81-LEN(J$) : PRINT J$;
1650 DX=(XMAX-XMIN)/10
1660 FOR I=0 TO 8 STEP 2          'lable x-axis loop
1670   J=XMIN+I*DX
1680   LOCATE 23,7+I*7 : PRINT USING "+###.#";J
1690 NEXT I
1700 '
1710 '********** MAIN PLOTTING SECTION ********************
1720 '
1730 SX = (XMAX - XMIN) / 560
1740 '
1750 FOR ML=1 TO 2
1760   FOR X = XMIN TO XMAX STEP SX
1770     IF ML=1 THEN GOSUB 8000 ELSE GOSUB 9000
1780     IF Y<YMIN OR Y>YMAX THEN 1820     'out of range, next value
1790     PY = 172 - (Y-YMIN) * 160 / (YMAX - YMIN)
1800     PX =  76 + (X-XMIN) * 560 / (XMAX - XMIN)
1810     IF X=XMIN THEN LINE (PX,PY)-(PX,PY) ELSE LINE -(PX,PY)
1820   NEXT X
1830 NEXT ML
1840 '
1850 IN$=INKEY$:IF IN$<>"" THEN GOTO 1850
1860 IN$=INKEY$:IF IN$= "" THEN GOTO 1860
1870 IF IN$<>"E" AND IN$<>"e" THEN 1850     'push E to exit loop
1880 END
1890 '
1900 '
7000 '========== PLACE CONSTANT VALUES HERE ===============
7010 '
7020 YMIN = -1.5     'minimum y value
7030 YMAX = 1.5      'maximum y value
7040 'place y-axis title below (18 characters, max.)
7050 Y$ = "THIS IS THE Y-AXIS"
7060 '
7070 XMIN = -360     'minimum x value
7080 XMAX = 360      'maximum x value
7090 'place x-axis lable below (60 characters, max.)
7100 X$ = "THIS IS THE GRAPH'S X-AXIS"
7110 '
7120 'place two title lines below (60 characters, max.)
7130 T1$ = "THIS IS THE FIRST TITLE LINE OF THE GRAPH"
7140 T2$ = "(And this is the second title line)"
7150 '
7160 RETURN
7170 '
7180 '
8000 '========== PLACE FUNCTION #1 BELOW ==================
8010 '
8020 Y = SIN(X*(PI/180))
8030 '
8040 RETURN
8050 '
8060 '
9000 '========== PLACE FUNCTION #2 BELOW ==================
9010 '
9020 Y = 1.2*SIN(X*(PI/180)+30*(PI/180))
9030 '
9040 RETURN
```

will also change other labels shown on the screen. We'll describe how to make those modifications a little later.

A moment or two after the title appears, if you've entered the program correctly, a graph similar to the one shown in Fig. 1 should materialize on your screen. The graph is a plot of two sinewaves having the same frequency, but different magnitudes and phase angles.

When the plotting process is complete, the program begins running a loop that is constantly looking for a press of the E key. Once E has been pressed, the program halts execution and returns you to the BASIC language editor (with the "OK" prompt displayed).

The Program. Having looked at an example of the type of chart Grapher is capable of producing, let's discuss how you can modify the program to graph the functions that you'd like to see displayed.

Let's analyze lines 7000–7140 of the program, looking at the sample graph shown in Fig. 1. Notice first of all that the variables YMIN and YMAX of the program, which are defined by lines 7020 and 7030, are used to specify the minimum and maximum values of the Y variable that will be displayed on the screen. In lines 7070 and 7080, XMIN and XMAX are defined in the same manner for use with the graph's X-axis. Simply modify the values (before you run the program) to accommodate the range of the variable you wish to plot.

Although any values for XMIN and XMAX will generally work, sometimes you may select values of YMIN and YMAX such that the points Grapher needs to plot are outside of the range displayed on the screen. The result of that is that no points will appear on the screen when the program is run. That would be the first place to look for mistakes if your graphs do not appear as you expect them to.

The variable Y$ in line 7050 is assigned a value to be used as the title of the graph's Y-axis, and the variable X$ of line 7100 is used for the title of the X-axis. Similarly, the program has two lines associated with the main title (7130 and 7140), and those lines assign values to variables T1$ and T2$.

To modify the four labels to fit your own graphs, merely change the string values to whatever text you would like to see on the graph. Because the program automatically centers the title and axis labels, there is no need to "pad" the strings with extra spaces. When entering the variables, be careful not to exceed the maximum allowable lengths; the Y-axis label can be up to 18 characters long, while the other three strings can be up to 60 characters in length.

The last thing you must do before Grapher can plot your functions is to type into the program the functions you want to plot. To do that, you need to enter each function as a subroutine, one starting at line 8000 and one at 9000. Your subroutines should always return a value in the variable Y for every value of X that is used by the routine; that is, every value of X that is between XMIN and XMAX.

In Listing 1, Function 1 is a simple sinewave with an amplitude of one and a phase angle of zero. Function 2 is also a sinewave, but with an amplitude of 1.2 and a 30° phase angle. Please keep in mind that BASIC assumes all angles are in radians. Because the variable X is in degrees, the ratio $\pi/180°$ is used to convert degrees into radians.

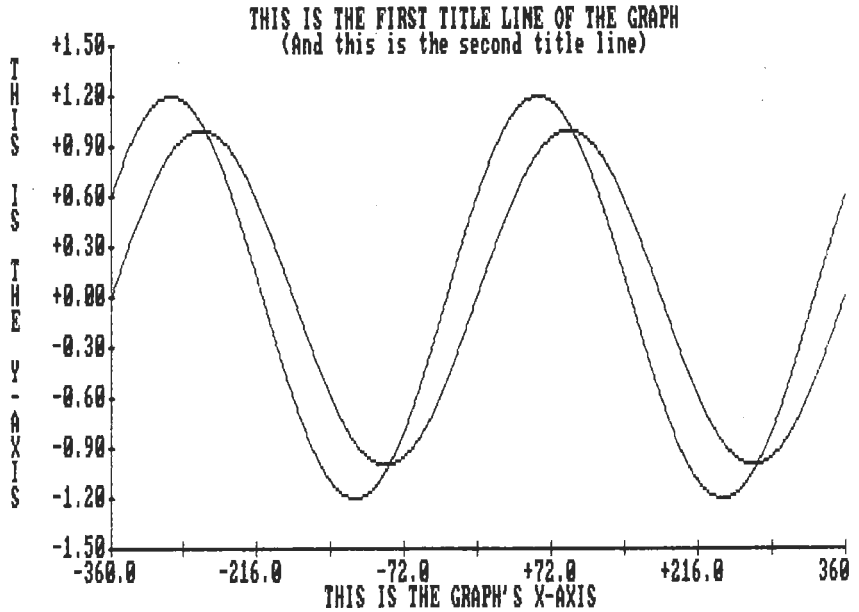

Fig. 1. The two sinewaves shown here are the "default" functions. To display your own functions and label the axis modify grapher as described in the text.

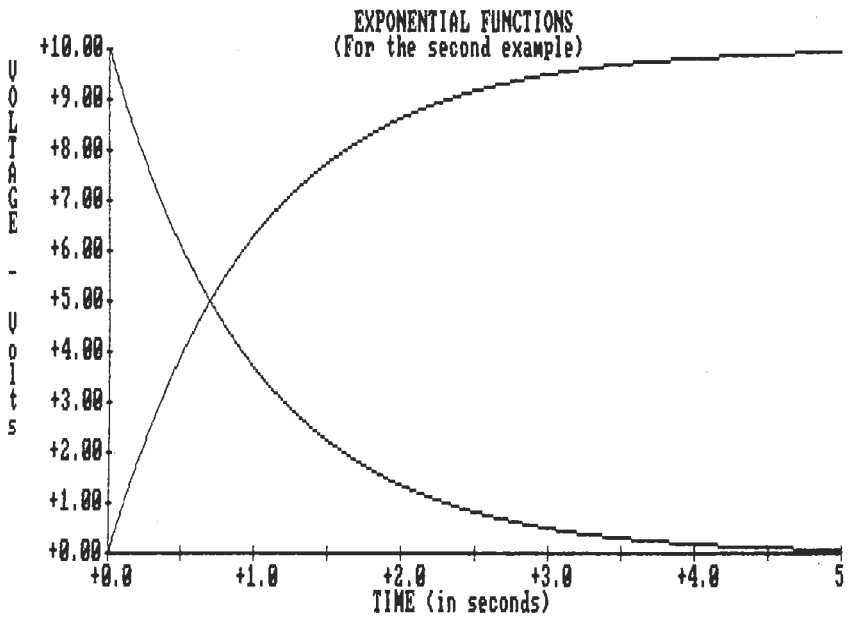

Fig. 2. This is another example of the Grapher program's output. Lines 7000–9040 of Grapher were modified as shown in Listing 2 on page 100 to allow the program to produce the graph shown in this illustration.

As an example of how to modify the program to use other functions, Fig. 2 shows another set of plots generated by Grapher: the rising and falling exponential functions. Listing 2 shows the modifications necessary to to generate that graph.

Keep in mind that the functions described by the subroutines don't have to be just one line long, they just have to return a single value of Y for every value of X in the range from XMIN to XMAX. For example, the subroutine listed below could be used to plot a 2-volt sinewave that has been clipped to +1.75 volts.

```
8020 Y= 2*SIN (X*(PI/80))
8024 IF Y>1.75 THEN Y=1.75
8028 IF Y<-1.75 THEN Y=-1.75
8040 RETURN
```

Another important point to remember is that subroutines must always end with a RETURN statement. Failure to include that statement, depending on where it is supposed to be located, is punishable by a simple error message, the halting of program execution, or the plotting of an incorrect graph.

Program Description. From Listing 1, it can be seen that Grapher is composed of three main sections of code and three "support" areas. The code sections are: 1) program initialization; 2) screen and variable initialization; 3) the main plotting section. The three support areas, which have already been discussed, are: 1) the constant values section; 2) Function 1 definition area; 3) Function 2 definition area.

A general description of each of the six segments of the program follows. For those readers who are more interested in a line-by-line description of how Grapher works, please refer to Table 1, which contains such a description.

Program initialization takes place in lines 1000–1340. This section starts by clearing the program's variables and displaying one screen of instructions. Then, in lines 1280–1320, the user is allowed to gracefully exit the program if they have not modified the print values in lines 7000–7160, or if they have not entered the functions they wish to plot as subroutines starting at lines 8000 and 9000.

In the next section of the program, from line 1350 to line 1700, both the program variables and the screen are initialized. This section takes care of drawing and labeling the graph's axes, displaying the titles of the axes, and displaying the title of the graph as a whole.

The last section of code, the part that handles the actual graphing of the functions, is contained in lines 1710–1900. An outer FOR-NEXT loop, which begins on line 1750, is used to step through each of the two functions in turn. An inner FOR-NEXT loop begins on line 1760 and its purpose is to step through each pixel (or "dot") on the X-axis from the minimum X value (XMIN) to the maximum value (XMAX).

More Words. Grapher's charts do not fill the entire PC screen, but rather use an area that is 160-pixels high by 560-pixels long, for a total of:

$$160 \times 560 = 89{,}6000 \text{ pixels}$$

Because the PC uses a method of displaying graphics known as bit-mapping (every pixel is represented by a

(Continued on page 100)