
Technical Notes

Plotter Drivers as an Exercise in Forth Wordset Design

A. Richard Miller and Thomas B. Dowling

*Miller Microcomputer Services
61 Lake Shore Road, Natick MA 01760-2099*

Abstract

Three levels of driver design are discussed for a typical X-Y plotter. A minimum driver merely permits Forth to mimic the plotter's internal BASIC command set. A more elegant one transforms the same commands into a more Forth-like and more English-like wordset. Finally, the command set is modified and vectored into the existing MMSFORTH TGRAPH (Turtle Graphics) utility. This transforms the plotter into an optional video screen to which one can output TGRAPH graphics routines without any modification. Source code for a simple Forth implementation for the Radio Shack FP-215 Flat-bed plotter is included, and can be adapted to many similar plotters.

Background

The intelligent computer plotter is an interesting and useful peripheral which can dramatically illustrate the different approaches to software design for peripherals in general. The Forth programmer has two obvious options: either he trains himself to talk (and think) like a plotter, or he trains the plotter to think like Forth. The second may seem the better to most of us. A third option, using Forth to train the plotter to merge invisibly into the application program, is the most useful for typical applications. Each of these three strategies is implemented in this paper, and the results may be easily demonstrated.

The bulk of our routines will be useable on many plotters, but they specifically match the Radio Shack TRS-80 FP-215 Flatbed Plotter. It is called a four-pen plotter and perhaps it is, in that four pens are neatly socketed in the front panel; however, it is up to the user to plug the right one in when appropriate! Happily, the other features of this plotter are a more than adequate set for use on any computer. And because the FP-215's internal firmware uses a command set which closely tracks that of many other plotters, our plotter drivers should lend themselves to many other applications.

Talking Down to the Plotter through Forth

The plotter connects to the computer's standard Centronics parallel port (or, optionally, serial) instead of a printer and uses a typical set of firmware commands which facilitate its programming in BASIC. Some of these are:

M x,y	MOVE (absolute) without drawing, to location x,y relative to the Origin.
R x,y	MOVE (relative) without drawing, from current pen location x steps horizontally and y vertically.
D x,y	DRAW (absolute) from current coordinate to destination x,y.

J x,y	DRAW (relative) from current pen location x steps horizontally and y vertically.
H	HOME moves to the Home position without drawing a line.
I x,y	Reset the ORIGIN without moving.
P characters	PRINT the string of alphanumeric characters in the current direction and size.
Q direction	Change the print direction: 0 # left-to-right, 1 # top-to-bottom, 2 # right-to-left, 3 # bottom-to-top (letters point left).
S size	SIZE of printed characters (height # size x 0.6mm; width # size x 0.4mm.).

Similar commands limit the plotting area, draw and mark axis lines, specify the pitch of dotted lines, and draw a variety of special symbols to mark data points, etc. All these commands would be sent to the plotter via the printer port as LPRINT commands in BASIC, such as LPRINT "M50,150", LPRINT "S4" and LPRINT "PSample printout."

The first level of plotter sophistication is to output through Forth in the same format. Using MMSFORTH's virtual I/O words PRINT and CRT, PRINT ." M50,150" ." S4" ." PSample print-out." CRT. A little better, but still in the flavor of the plotter's commands and that confusing Perverse Algebraic Notation (PAIN).

Teaching the Plotter to Listen to Forth

Instead of learning how to talk down to this plotter, we use Forth to teach it to listen to us. Blocks 136 and 137 do this, by establishing CREATE..DOES> constructs (LINES=, 1OUT=, 2OUT=) for defining new Forth words and then building one such word for each of the plotter commands. This time we can shape them to take parameters from the Forth stack, can give them names as long or cryptic as we wish, etc. Thus M becomes MOVE, R becomes RMOVE, D becomes DRAW, J becomes RDRAW, etc. Only one unusual word is here, PHOME for PlotterHOME, because later we will be adding this to another wordset which already has a simple HOME we wish to maintain.

Using this small and instantly loading wordset, we can take over the plotter in true Forth. From there, we can issue instructions in a more English-like mode, and also can build more complex Forth words using this set. The simple application, with source code in Blocks 53 and 54, plots Figure 1.

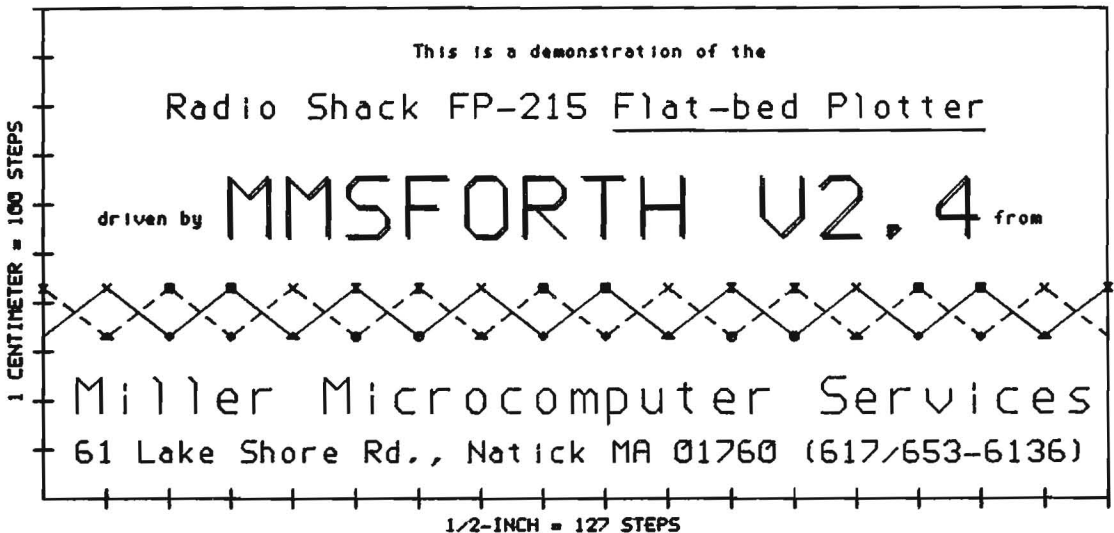


Figure 1.

Vector a few Plotter-Translating Routines into your Application Program

The great failing of most plotter drivers is that they are quite incompatible with video drivers. Thus a fine program to display a picture cannot be reduced to the vectored-lines information wanted by a plotter. A common compromise is to dump the picture to a dot-matrix printer by copying the settings of the individual raster dots (pixels). MMSFORTH's TGRAPH (Turtle Graphics) utility includes a screen dump utility for several popular printers, which has been used for many professional applications.

But the internal calculations of TGRAPH have far better resolution than the video display, and so does the plotter. And like most good Forth programs, TGRAPH has neatly factored out the critical routines down low, where a few changed routines can still be called by all the same higher level words. So we change them, by "subroutining" a different routine into these few words to adjust them for the plotter instead of the screen. There are various ways to accomplish this; MMSFORTH provides an efficient VECT construct which is ideal. We prepare by vectoring one critical word in the existing TGRAPH source code, DRAW.LINE, and adding one more, M.PT. DRAW.LINE was the way to connect two points, but now it will be only one of the ways to do so (the one for video display). So we define a VECT D.LINE to be the new generic routine, and initially assign the existing DRAW.LINE subroutine to it with [FIND] DRAW.LINE IS D.LINE. All subsequent program occurrences of DRAW.LINE are edited to D.LINE. In Block 138, we define a plotter version of this code as X.LINE, and immediately follow with a definition of MV1 to have the plotter move to a point. There is no equivalent routine for video, so we also define a VECT M.PT on Block 164 and initially assign no action to it with [FIND] NULL IS M.PT. With these "ambidextrous" VECT words installed, Block 138 also defines switching words, PLOTTER and SCREEN, to change gears on command. PLOTTER does a [FIND] X.LINE IS D.LINE [FIND] MV1 IS M.PT to switch these words into action, while SCREEN sets them back to the video routine with [FIND] DRAW.LINE IS D.LINE [FIND] NULL IS M.PT

For those unfamiliar with MMSFORTH V2.4, our [FIND] is a state-smart version of FIND which works like our state-smart `'`, except it returns the CFA instead of the PFA. Other trickery in these switching words does associated housekeeping, maintaining appropriate ratios between the zoom and horizontal/vertical sizings on screen and plotter (so a full picture on one will maintain proper size and aspect ratio on the other), hiding the turtle and ignoring varying foreground and background colors on the plotter (however they are displayed on the video), etc.

Using this most sophisticated level shows its ultimate power. A display is optimized on screen and, when ready, vectored to the plotter with a simple PLOTTER command and a repeat of the display drawing Forth routine. In our demonstration on an IBM personal computer, we also use MMSFORTH's print spooler, sized to 64K RAM. This minimizes computer down time on long plots, while taking advantage of the improved resolution of TGRAPH and the plotter mechanism.*

Credits & References

The attached blocks of plotter code were written by Tom Dowling of Miller Microcomputer Services. They are copyright 1985 by MMS; noncommercial use is permitted provided that proper credit is given.

The MMSFORTH TGRAPH utility was written by Tom Dowling of MMS, and is available under license. It has been described in a paper, "An Implementation of High-Resolution Graphics (Including Turtle Graphics) in MMSFORTH", by A. Richard Miller and Thomas B. Dowling (1982 FORML Conf. Proc., pp. 281-82).

The source code makes references to several advanced Forth constructs. VECT words are addressed with IS, as is this CONSTANT-type usage of our (three-CFA) QUAN words. TGRAPH implements a similar TO word (after the older TO-VARIABLE). Papers on these subjects include:

"The 'TO' Solution" by Paul Bartholdi, *FORTH Dimensions* 1:4, Dec. 1978, pp. 38-40.

"High Speed, Low Memory Consumption Structures" by Evan Rosen, *1982 FORML Conf. Proc.*, pp. 191-96.

"The QUAN Concept Expanded" by Tom Dowling, *1983 Rochester Forth Conf. Proc.*, pp. 89-92; also *Journal of Forth Applic. & Rsch.* 1:2, Dec. 1983, pp.69-72.

The special LOGO issue of *BYTE Magazine* (August 1982) provided many classic Turtle Graphics patterns. They run equally well and much faster in MMSFORTH and TGRAPH. We particularly like the extravagant DOILY pattern (Figure 2) which is included with TGRAPH. Christopher Keavney originally put it on the MIT Student Info. Proc. Board; Glenn Forester resolved that pattern into a LOGO routine based on pentagonal spirals, as reported in "Problem-Solving with LOGO: Using Turtle Graphics to Redraw a Design" by William Weinreb, *BYTE*, Nov. 1982, pp. 118-34.

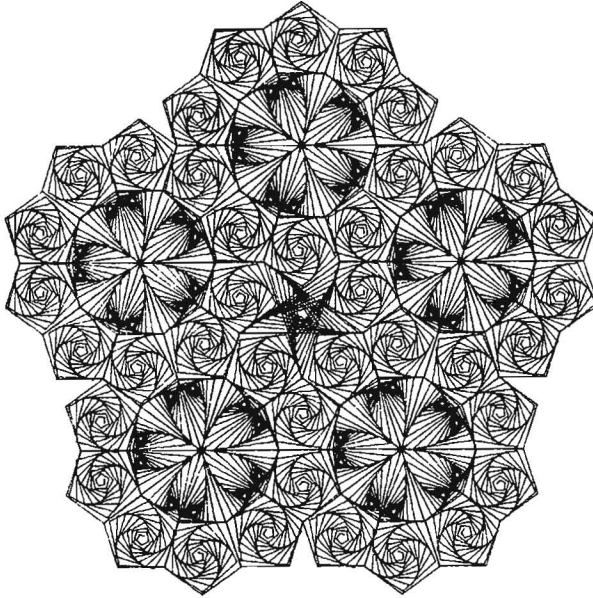


Figure 2.

Manuscript received June 1985.

Dick Miller is a partner in Miller Microcomputer Services, developer of state-of-the-art microcomputer software at all levels from operating system through language (MMSFORTH) to such varied application programs as word processors, database systems, and communications modules. Before founding MMS in 1977, Dick worked as an electrooptical physicist for Polarad Electronics Corp., Block Engineering, and EG&G, Inc. Dick holds a BS in Physics from Queens College and has studied at Rensselaer Polytechnic Institute, Brooklyn Polytechnic Institute and New York City's Hayden Planetarium.

Tom Dowling has been the lead programmer for Miller Microcomputer Services since 1979, and has headed many major projects using Forth and other computer languages. He has a BS in Electrical Engineering from Pratt Institute and a MS in Electrical Engineering from Northeastern University.

BLOCK 331

```

0 ( 850401mms RS/FP-215 PLOTTER DRIVER, 1 of 2 )
1
2 : P-PAIR      SWAP 0 .R ." 0 .R CR CRT ;
3 : P-$        COUNT PRINT ." P" TYPE CR CRT ;
4 : PLOT-IT    R> DUP C@ 1+ OVER + >R P-$ ;
5
6 : P" STATE C@ IF    COMPILE PLOT-IT ,"$"
7                   ELSE 34 WORD P-$ THEN ; IMMEDIATE
8
9 : LINES= CREATE BL WORD 1+ C@ C, DOES> PRINT C@ EMIT P-PAIR ;
10
11 LINES= DRAW D    LINES= MOVE M
12 LINES= RDRAW J   LINES= RMOVE R    LINES= ORIGIN I
13
14 : V-AXIS PRINT ." X0," P-PAIR ;
15 : H-AXIS PRINT ." X1," P-PAIR ;
-->

```

BLOCK 332

```

0 ( 850401mms RS/FP-215 PLOTTER DRIVER, 2 of 2 )
1
2 : 1OUT= CREATE , BL WORD 1+ C@ C,          DOES>
3 PRINT DUP @ SWAP 2+ C@ EMIT SWAP ABS SWAP MOD 0 .R CR CRT ;
4
5 6 1OUT= SYMBOL N          4 1OUT=          DIRECTION Q
6 256 1OUT= SIZE S          128 1OUT= DOTTED-LINE-PITCH B
7
8 : 2OUT= CREATE BL WORD 1+ @ ,
9 DOES> PRINT DUP C@ EMIT 1+ C@ EMIT CR CRT ;
10
11 2OUT= SMALL-PLOT F0      2OUT= FULL-PLOT F1
12 2OUT= SOLID-LINE L0      2OUT= DOTTED-LINE L1      2OUT= PHOME H
13
14 : INITIALIZE-PLOTTER SMALL-PLOT SOLID-LINE 4 SIZE 0 DIRECTION
15 30 DOTTED-LINE-PITCH 0 0 ORIGIN PHOME ;
-->

```

BLOCK 333

```

0 ( 850605mms TGRAPH>PLOTTER routine )
1
2 : X.LINE ( x1 y1 x2 y2 gtype -> )
3 0= IF 2SWAP 2DUP CUR-POINT XPT D= IF 2DROP ELSE NEGATE MOVE
4 THEN NEGATE DRAW ELSE 2DROP 2DROP THEN ;
5
6 : MV1 2DUP XPT NEGATE MOVE ;
7
8 : PLOTTER [FIND] X.LINE IS D.LINE [FIND] MV1 IS M.PT
9 75 IS %RATIO %SIZE %SIZE 100 TO YSCALE 1350 930 ORIGIN
10 0 TO XOFFSET 0 TO YOFFSET 0 TO GTYPE INIT HIDETURTLE ;
11
12 : SCREEN [FIND] DRAW.LINE IS D.LINE [FIND] NULL IS M.PT
13 X% 4 = IF 88 10 ELSE 44 20 THEN IS %RATIO TO YSCALE
14 %SIZE %SIZE 128 TO XOFFSET 100 TO YOFFSET 0 TO GTYPE INIT ;
15 SCREEN ( Enter PLOTTER to shunt output to plotter. )

```

BLOCK 248

```

0 (850411mms RS/FP-215 PLOTTER DEMO, 1 of 2 )           : TASK ;
1 INITIALIZE-PLOTTER
2 ( Print text )                                         4 SIZE
3 960 1550 MOVE P" This is a demonstration of the"      8 SIZE
4 460 1425 MOVE P" Radio Shack FP-215 Flat-bed Plotter"  4 SIZE
5 320 1210 MOVE P" driven by "           0 -25 RMOVE    20 SIZE
6                                     P" MMSFORTH V2.4       0 25 RMOVE    4 SIZE
7                                     P" from"                12 SIZE
8 275 825 MOVE P" Miller Microcomputer Services"        7 SIZE
9 275 725 MOVE P" 61 Lake Shore Rd., Natick MA 01760    (617/653-
10 6136)"
11 ( Emphasize text )
12 1370 1400 MOVE 750 0 DRAW                               4 SIZE
13 325 1180 MOVE P"                                       20 SIZE
14                                     P" MMSFORTH V2.4"      4 SIZE
15                                                         -->

```

BLOCK 249

```

0 ( 850411mms RS/FP-215 PLOTTER DEMO, 2 of 2 )
1 ( Frame text )
2 207 650 MOVE 100 10 V-AXIS 207 650 MOVE 128 17 H-AXIS
3 0 1000 RDRAW -2176 0 DRAW
4 ( Create ZIGZAG routine, like line graph ) 17 DOTTED-LINE-PITCH
5 QUAN Y 100 IS Y ( vertical motion for ZIGZAG )
6 : MARK ( I' -> ) 6 MOD SYMBOL ; ( vary the symbol )
7 : ZIG ( X -> X ) DUP Y DUP NEGATE IS Y RDRAW I' MARK ;
8 : ZIGZAG SOLID-LINE 128 17 0 DO ZIG LOOP DROP DOTTED-LINE
9 0 -100 RMOVE Y NEGATE IS Y -128 17 0 DO ZIG LOOP DROP ;
10 207 980 MOVE ZIGZAG
11 ( Label the axes; can use to measure future plotter spacings )
12 162 850 MOVE 3 DIRECTION P" 1 CENTIMETER = 100 STEPS"
13 1025 585 MOVE 0 DIRECTION P" 1/2-INCH = 128 STEPS"
14
15 PHOME ( Return pen to lower right ) FORGET TASK

```