TEKLIB for CP/M 2.2

Martin Hepperle, May 2020

TEKLIB is a small library of basic functions to generate graphics output for Tektronix terminals (or emulators) from Fortran programs. I wrote it around 2018, before I discovered the GSX-80 system.

Using TEKLIB

The library was written for a CP/M 2.2 system using Microsoft Fortran 3.44. It can be linked to your main Fortran program with the Microsoft L80 linker.

The library provides the functions listed below.

A program must call TINIT first and must terminate the library usage with TCLOSE.

Between these brackets you would use TVIEW and TWIND to set up your user coordinate system and then use the rest of the routines.

File	Routine	Parameters	
tarc.for	TARC(X,Y,R,PHI0,PHI1,DPHI) Draw an arc with given center point.	REAL X, Y REAL R REAL PHI0 REAL PHI1 REAL DPHI	center point radius start angle end angle angular step
taxis.for	TXAXIS(X0,X1,YPOS,XSTEP,YTICK) Draw a horizontal axis with tick marks.	REAL X0 REAL X1 REAL YPOS REAL XSTEP REAL YTICK	start point end point intercept with the y-axis tick spacing tick length
taxis.for	TYAXIS(Y0,Y1,XPOS,YSTEP,XTICK) Draw a vertical axis with tick marks.	REAL Y0 REAL Y1 REAL XPOS REAL YSTEP REAL XTICK	start point end point intercept with the y-axis tick spacing tick length
tchar.for	TCHAR(X,Y,ICHR,SIZE,ANGLE) Draw a vector graphics character.	REAL X REAL Y INTEGER ICHR REAL SIZE REAL ANGLE	horizontal position vertical position ASCII code number text height in device units baseline angle in degrees
tchar.for	TSFONT(IFONT) Select a font. Font #1 is the default, font #2 has greek characters. Fonts are loaded from files FSIMPLEX.FNT or FGREEK.FNT.	INTEGER IFONT	font number, 1 or 2
tclear.for	TCLEAR Clear the screen.		
tclose.for	TCLOSE Close the output device.		
tdraw.for	TDRAW(X,Y) Draw a line from the current point.	REAL X REAL Y	horizontal position vertical position
tdrawc.for	TDRAWC(X2,Y2) Draw a line from the current point – with clipping.	REAL X REAL Y	horizontal position vertical position
tidraw.for	TIDRAW(IX,IY) Draw a line to the point in device units.	INTEGER IX INTEGER IY	horizontal position vertical position
timove.for	TIMOVE(IX,IY) Move the current point in device units.	INTEGER IX INTEGER IY	horizontal position vertical position
tinit.for	TINIT(NDEV)	INTEGER NDEV	unit number for output

	This initializes the library and opens the given Fortran unit number. Microsoft Fortran 3.44 connects the units 1, 3, 5 to the console (TTY:), 2 to LST: and, 4 to the PUN: device.		
tmove.for	TMOVE(X,Y) Move the current point.	REAL X	horizontal position
		REAL Y	vertical position
tmovec.for	TMOVEC(X,Y) Move the current point– with clipping.	REAL X	horizontal position
		REAL Y	vertical position
tnum.for	TNUM (X, Y, SIZE, ANGLE, VAL, NDIG, NALIGN) Draw a number at the given position.	REAL X	horizontal position
		REAL Y	vertical position
		REAL SIZE	text size in device units
		REAL ANGLE	baseline angle in degrees
		REAL VAL	number to output
		INTEGER NDIG	decimal digits
		INTEGER NALIG	alignment flag
trect.for	TRECT(X0,Y0,X1,Y1) Draw a rectangle given by its diagonal points.	REAL X0,Y0	first corner point
		REAL X1,Y1	second diagonal point
ttext.for	TTEXT(X,Y,TEXT,LEN,SIZE,ANGLE) Draw a text string at the given position.	REAL X,Y	starting point
		INTEGER*1 TEXT(255)	string to write
		INTEGER LEN	number of characters
		REAL SIZE	text size in device units
		REAL ANGLE	baseline angle
tview.for	TVIEW(XLO,XHI,YLO,YHI) Define a viewport relative to the paper bounds.	REAL XLO	left border rel. to paper
		REAL XHI	right border rel. to paper
		REAL YLO	low border rel. to paper
		REAL YHI	high border rel. to paper
twind.for	TWIND(XLO,XHI,YLO,YHI) Define a user coordinate (UC) system in the current viewport.	REAL XLO	UC for left border
		REAL XHI	UC for right border
		REAL YLO	UC for low border
		REAL YHI	UC for high border
taxlab.for	TXALAB(X0,X1,YPOS,XSTEP,SIZE,ANGLE,NDIG) Label a horizontal axis.	REAL X0	start point
		REAL X1	end point
		REAL YPOS	intercept with the y-axis
		REAL XSTEP	label spacing
		REAL SIZE	text height in device units
		REAL ANGLE	baseline angle in degrees
		INTEGER NDIG	number of decimals
taxlab.for	TYALAB(Y0,Y1,XPOS,YSTEP,SIZE,ANGLE,NDIG) Label a vertical axis.	REAL Y0	start point
		REAL Y1	end point
		REAL XPOS	intercept with the y-axis
		REAL YSTEP	label spacing
		REAL SIZE	text height in device units
		REAL ANGLE	baseline angle in degrees
		INTEGER NDIG	number of decimals
trast.for	TXRAST(X0,X1,XSTEP,Y0,Y1) Draw a set of vertical raster lines moving along the horizontal direction.	REAL X0	start point of raster
		REAL X1	end point of raster
		REAL XSTEP	spacing
		REAL Y0	start point of each line
		REAL Y1	end point of each line
trast.for	TYRAST(Y0,Y1,YSTEP,X0,X1) Draw a set of horizontal raster lines moving along the vertical axis.	REAL Y0	start point of raster
		REAL Y1	end point of raster
		REAL YSTEP	spacing
		REAL X0	start point of each line
		REAL X1	end point of each line

If not specified otherwise, all values are in user coordinate system units.

Portability

Because early Fortran standards do not have character or string types, TEKLIB uses the INTEGER*1 type to handle characters and bytes. Some Fortran implementations provide a non-standard BYTE type but only with Fortran-77 the CHARACTER type became available. You might have to replace the type definitions accordingly.

About 99% of the library is written in Fortran. There is one small assembler routine which outputs a buffer of bytes to a CP/M output device. This routine circumvents Fortran-specific carriage control behavior. Depending on the target system and the Fortran implementation, this routine can be replaced e.g. by output to a binary direct access file, by a FORMAT with a leading “1H+” carriage control character or by another assembler routine.

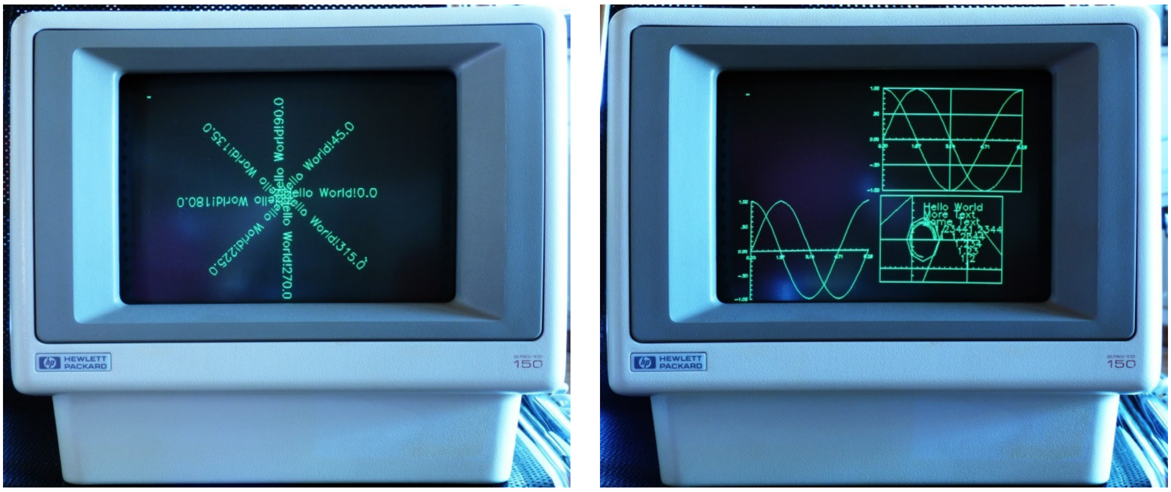

Figure 1: Output of the TEK1 and TEK2 test programs on a HP 150 with Tektronix emulation.