

The Elliott 900 QPLOT package.

Recording work done in 2012.

Terry Froggatt, January 2013.

Introduction.

In 2012, I refurbished a brace of Benson-Lehner incremental plotters which had been interfaced to Elliott 903 computers. Although I've had copies of the software to drive these plotters since working for Elliotts in the 1960s & 1970s, I'd never previously had access to any plotters.

So I'd never subjected the plotter software to the sort of improvement that I'd undertaken in the past, on other software issued with 903 and from elsewhere, when I developed the 900-series libraries for Airborne Computing Division and Maritime Aircraft Systems Division.

In understanding the existing plotter software, I found several problems, and I also realised that the code could be made both somewhat shorter and easier to understand. This report gives a summary of the work that I did in 2012.

Available Documentation.

Grey binder "Elliott 900 Manual"

Volume 1 : Functional Specification, Copy No. 1A/259

Part 4 : Peripheral Equipment

Section 3 : Digital Plotter DPA 381. April, 1968. (Issue 3).

This describes the plotter itself. It describes pen movements as East-West and in the Y-axis, and spindle movements as North-South and in the X-axis.

Grey binder "Elliott 900 Manual"

Volume 2A : Programming Information, Manual Copy No: 259

Part 1 : Programming Languages

Section 2 : Algol

Appendix 3 : Use of Optional Peripherals. (Issue 4).

This describes the Algol plotter package. It describes pen movements as East-West, and spindle movements as North-South, but which is the X-axis and which is the Y-axis depends on SETORIGIN's PAGE parameter. I've not seen a Fortran plotter package.

Grey binder "Elliott 900 Manual"

Volume 2C : Programming Information, Manual Copy No: 259

Part 4 : Peripheral Routines

Section 2 : QPLOT. (Issue 2).

This describes the SIR plotter package. It contains an additional "page 6a" (Issue 2), "2.6 Initial Entry to QPLOT", not in the contents list, which I shall need to refer to again further on. It describes pen movements as in the X-axis (limited by plotter width), and spindle movements as in the Y-axis. It says that a line from (X=0,Y=0) to (X=10,Y=1) is drawn as 9 "North" increments and 1 "North East" increment. However, Newsletter 8 for Volume 2, December 1967, says that (X=10,Y=1) should be changed to (X=1,Y=10) in this example. (It also corrects "QPLOT+4 and QPLOT+5", which appears just below the example, to "QPLOT+5 and QPLOT+6").

Grey binder "Elliott 900 Manual"

Volume 3 : Operating Instructions

Part 4 : Peripheral Equipment

Section 3 : Digital Plotter. (Issue 2A).

This describes the use of the plotter and test program X52. It describes pen movements as East-West, and spindle movements as North-South. (I found this section at the back of a binder which was spine-labelled Volume 3 parts 1 & 2, Copy No. 3A/84).

Grey binder "Elliott 900 Manual"

Volume 4 : Engineering Maintenance

Part 4 : Peripheral Equipment, Copy No. 4E/266

Section 3 : Digital Plotter (DPA 381). August, 1967. (Issue 2).

This simply says that the X52 plotter test program should be run every six weeks, and for a technical description of the plotter, see "the manufacturer's handbook".

The handbook is presumably Benson-Lehner's "Digital Incremental Plotter Operation & Maintenance Manual" which I also do have. It describes pen movements as in the Y-axis, and spindle movements as in the X-axis.

I also believe that I now have a complete set of large loose circuit diagrams, both for the generic plotter, and for the 903-specific interface fitted to both refurbished plotters. 903-specific plotters use different power-supply voltages to those shown in the generic design.

Axis System Analysis.

Benson-Lehner's manual and the Elliott Functional Specification are consistent in that they describe pen movements as in the Y-axis and spindle movements as in the X-axis. This corresponds to the view that the plotter will be used as a "chart reorder", plotting a variable across the page in the Y-axis as the paper is steadily rolled out over an arbitrary length of time in the X-axis, and that this will be best watched from the right-hand side of the plotter.

The SIR implementation takes the opposite view that the plot is a diagram, probably of predetermined size, to be drawn for viewing from the front of the plotter, with pen movements in the X-axis and spindle movements in the Y-axis. The author of the QPLOT example appears to have been unaware of this inconsistency, and so initially got the interpretation of North-South and East-West wrong. But with the correction, these compass terms *are* used consistently.

The Algol PAGE parameter allows either view to be selected. The joystick on the plotter itself is just marked with arrows, and so avoids all of the above terminology.

Available Tapes.

My main starting point was the SIR subroutine source tape "903 QPLOT ISSUE 1". I have two copies of this, issued with copies 259 & 502 of the 903 library respectively. They differ in just one bit. Both have legible headers. In copy 502 the whole tape is even parity, including the legible header, whereas copy 259 has one odd parity row in the legible header. So I've taken "903 QPLOT ISSUE 1 copy 502" as definitive. The tape image is included in the ZIP file as

QPLOT502.RAW. Both copies of the SIR source contained a spurious "h" between the legible header and the SIR subroutine code.

Also in the ZIP file is QPLOT502.SIR, which is an ASCII version of QPLOT502.RAW, readable with any text editor. In this, the parity track, blanks & erases, the legible header, and the spurious "h", have all been removed. The layout has been re-arranged into the usual columns, 6 characters wide for the instruction function and 10 characters wide for the other columns, and multiple blank lines have been reduced to single blank lines. However, I've *not* changed any of the printable text.

The other starting point was 903 Algol Tape 3, which is the Algol Library Tape in Relocatable Binary form. I had several copies of this, some with legible headers and some without, variously at Issue 5, Issue 6, & Issue 7. I did not have the SIR source files that would have been used to create these library tapes, so I wrote a program to convert the relocatable binary into a readable form, mainly using block-relative addressing akin to 903 T2. This confirmed that I really only had one version of each issue, once the legible headers and insignificant blanks were ignored. The versions of the Plotter package in Issues 5 & 6 were found to be identical, except that each used the absolute addresses appropriate for the corresponding issue of the Algol interpreter. Issue 7 contained no plotter package.

The next task was to recover proper SIR source files for the Algol library plotter package. It was clear, from the decoded T2 version, that much of it was the same as the SIR QPLOT, so I only had to invent identifiers for a small amount of Algol interfacing code. Having done this, I assembled the results with Load-&-Go SIR to check that I got the correct relocatable binary tapes again. (Forward jumps to labels & forward relative jumps generate different relocatable tapes, even though they result in the same store contents). The Algol plotter package is in two parts, so that programs needing just line-drawing can avoid loading the character-drawing code. These two parts for issue 6 are in the ZIP, as QAPLT1_6.SIR & QAPLT2_6.SIR. The two parts for issue 5, QAPLT1_5.SIR & QAPLT2_5.SIR, are identical except that "BASE=100" is replaced by "BASE=0" in each of them.

The 15/6/12 releases of QPLOT & QPLOTTEST.

In understanding the existing plotter software, I found several problems, and I also realised that the code could be made both somewhat shorter and easier to understand.

After giving an **error message** on the TeleType, QPLOT ISSUE 1 waited to be retriggered at location 9, which it did by planting a jump instruction into location 9, thereby corrupting it. Although Newsletter 8 for Manual Volume 2, December 1967, says that "the 903 user should not normally alter or place program in locations 8 to 20", it already then was (and still is) quite normal to locate user programs from location 8 onwards. Also, the QPLOT ISSUE 1 retriggering code assumes that QPLOT is in the first store module, and that it is OK to continue on top level. In my 15/6/12 version, after giving the error message on the TeleType (and sounding its bell using a spare half-word), I simply wait for any character to be input from it. The Algol plotter package does not give error any indications; unimplemented output characters are replaced by "^" and the illegal parameter error cannot occur.

The QPLOT ISSUE 1 line-drawing routine explicitly checks for **lines of zero length**. This test is performed if the pen is to be raised or lowered before drawing the line, but not if the pen is to be left at its present height (when the pen will be stepped one unrecorded increment to the East).

There is perhaps some logic here, in that users might want to raise or lower the pen without making any X or Y movements, but if they don't want to raise or lower it OR make any X or Y movements, then they won't call QPLOT. However, a reasonable way to draw a curve with QPLOT is to lower the pen, then calculate a sequence of points (parametrically) no more than an increment apart and often less, and call QPLOT to draw short line segments between them. If this is done with ISSUE 1, the pen will drift steadily across the paper to the right. In my 15/6/12 version, I've moved the exit test of the main stepping loop from the end of the loop to the start, which cures the problem and removes the need for the initial 8-word test for lines of zero length completely. The Algol plotter package is OK: it explicitly instructs QPLOT to raise the pen for MOVEPEN and lower it for DRAWLINE.

The time taken for the plotter to raise or lower the pen is six times the time taken to move one increment in any of the eight compass directions, and cannot reliably be done at the same time. So it is sensible to avoid unnecessary **pen up & down movements**. The QPLOT ISSUE 1 line-drawing routine does this in two ways. Firstly there is a parameter for the caller to select between raising or lowering the pen or doing neither, before any X or Y movement. The QPLOT character-drawing routine is careful to use this, selecting pen down for the first stroke and neither for the remaining strokes (except within "=" and "10"). Sadly it moves from the starting position to the first stroke without selecting pen up, and so can produce a spurious line. Secondly QPLOT attempts to suppress explicit up & down requests which appear to be unnecessary. This is important when using Algol to draw curves, because DRAWLINE always instructs QPLOT to lower the pen, which could make curve drawing very slow. There is no way of reading the pen position back from the plotter, so QPLOT keeps track internally of whether the pen is up or down, which raises the question of how the internal flag is initialised. Algol's SETORIGIN raises the pen directly (without calling QPLOT) and sets QPLOT's internal flag accordingly. Unfortunately the SIR programmer has no access to the flag, which is why "page 6a" (added as an afterthought to manual section 2.4.2, see above) suggests initialising the plotter by calling QPLOT twice, once to lower the pen and once to raise it. Whatever the initial state of the flag, QPLOT will not suppress both movements. After this the flag and pen will agree (but there might be a dot on the paper). In my 15/6/12 version, the QPLOT line-drawing routine still only lowers the pen if the flag shows that it is not already lowered, so curve-drawing is still efficient, but it always raises the pen when told to do so, regardless of the flag, so this can be used to synchronise the flag and pen (without marking the paper). Relative to the original, my version will output an occasional extra pen-up command to the plotter, for example between two characters (which now correctly start with a pen-up movement from, and end with a pen-up movement to, the reference point). I'm assuming that curves are rarely drawn with the pen up: this is now slow. In my 15/6/12 version, the QPLOT character-drawing routine now leaves it to the line-drawing routine to optimise pen up & down movements. The "do neither" parameter is effectively redundant.

The QPLOT ISSUE 1 character-drawing routine has some variables which are initialised immediately before exit, so that they are right for any subsequent call of QPLOT. They are **not initialised** before the first call, although they will have been preset when the program was loaded into store. So, if a program using the character-drawing routine is stopped whilst still running, it cannot safely be restarted from the start without reloading it. It is quite likely that the plotter will be used to record the results of a real-time experiment in an endless loop, which will be stopped at the end of each experiment, with the expectation that another experiment can be started without reloading the program. This error also affects the Algol plotter package. It takes no more code to perform the initialisations correctly at the start of the routine than at the end, which I do in my 15/6/12 version.

The QPLOT ISSUE 1 character-drawing routine has a parameter, which is positive to draw characters along a line, and negative to draw a character centred on a point. The absolute value of the parameter determines the size of the character, but the interpretation of this size differs: for any given value, centred characters are drawn **twice as large** as linear characters. This is presumably an error. In Algol, the WAY procedure enables the user to control the size of linear characters in "print" statements, but the user is unaware of the size used by the CENCHARACTER procedure for the centred "+", and so is unaware of the difference. To retain compatibility with ISSUE 1, I have *not* corrected this error in my 15/6/12 version. The Manual states that the SIR parameter is an even number (but this is not checked) in the recommended range 12 to 100. Algol recommends a range 4 to 25 for WAY's parameter, which it multiplies by 5 before calling QPLOT. 25*5 is not even, and CENCHARACTER calls QPLOT with a size parameter of -15.

Whilst coding style is a matter of personal preference, I would expect to see an if-then-else construct coded into SIR using an unconditional forward jump at the end of the then-part around the else-part (if there is no good reason to do otherwise), with the use of backwards jumps confined to the ends of loops. The code in QPLOT ISSUE 1 generally places the else-parts some distance away from the then-part, and often reunites the parts with a backward jump. I think it is fair to call this **spaghetti coding**. Amongst other observations, three further oddities are worth specific mention. At one point there are two consecutive unconditional jump instructions, meaning that the second ("L23 8 L16") could be eliminated (simply by changing L23 elsewhere to L16). One of the calls of the ERROR routine is conditional, and so it executes its parameter (a data item) if there is no error. The ERROR routine itself uses a "live" instruction, suggesting that the author did not understand how to use B-modification.

There are three major sources of **inefficiency** in QPLOT ISSUE 1, which increase the code size (although they have no impact on speed since the plotter itself is the limiting factor).

Firstly, the logic at the start of the line-drawing routine branches into eight separate octants, with very little shared code, whereas the same effect can be achieved using just three if-then-else sections of code following one another.

Secondly, the main loop in the line-drawing routine uses two multiplications and a double-length subtraction to calculate the "running discriminant", $(2D+1)*L/2-C*M$ (and it sets the variable B+1 even though this is never used). If the first calculation is performed outside the loop, where C=1 and D=0, it becomes trivially $L/2-M$. Subsequently, the discriminant can be kept correct by adding L instead of incrementing D, and by subtracting M when C is incremented each time round the loop (which can be combined with the initial subtraction), using single-length arithmetic. This is the normal implementation of Bresenham's algorithm.

Finally the coding of the character-drawing routine branches six ways to separate the "up" and "end" flags from the Y-coordinate. Whilst it is odd that the flag bits are added to the absolute value of the coordinate, they can be separated quite simply despite this.

I have allowed myself one extra facility in the 15/6/12 version.

I have added a flag, SENSE, accessible globally as QPLOT-1, set to +1 by default:

- when SENSE = +1, the X & Y axes are as in QPLOT ISSUE 1,
- when SENSE = +0, the X & Y axes use the Benson-Lehner convention,
- other values of SENSE *must not be used*.

If the Manual's "page 6a" code is to be used to initialise the plotter, SENSE should be set to +1 whilst this is done. Although this facility is implemented within the line-drawing routine, its main

benefit is that it enables the character-drawing routine to write either **up or across** the page. This provides the same facility in SIR as provided by SETORIGIN's PAGE parameter in Algol. (Implementing this in the SIR routine needed only the flag and three extra instructions. It greatly simplifies implementation of Algol's PAGE, which previously overwrote an 8-word compass-point table in QPLOT which I've eliminated, by one of two other 8-word tables.)

My 15/6/12 version of QPLOT corrects the problems noted above, except that I have avoided changing the scaling of characters. It should be an **exact replacement** for QPLOT ISSUE 1, except for taking less store. The character table format is still as described in the Manual, and the user is free to alter the table. The 15/6/12 version of QPLOT was the one used that I used in the test program QPLOTTEST also dated 15/6/12, which demonstrates printing characters in both axes and then draws a spiral.

The 4/11/12 releases of QPLOT & QAPLT.

The character vertex table in QPLOT ISSUE 1 use one word per vertex, to store an X-coordinate which the Manual says is between -7 & $+7$, a Y-coordinate between -9 & $+9$, an "up" bit and an "end" bit. In my 4/11/12 version, I've packed the coordinates of **two vertices** into each word, which leaves no bits for the "up" and "end" flags. The end of each character is found using the address of the subsequent character in the pointer table, on the assumption that the characters appear in the vertex table in internal code order (which is not the case in ISSUE 1, but is easily arranged at no cost). A separate extra "up" half-word now is required (with $Y = -16$) to indicate when the pen should be raised between vertices (which only occurs in "=" and "₁₀"), and this same half-word can be used to fill the final word of a character. Details are given in comments near the tables in the SIR source code.

The characters "*" and "₁₀" were found to have X-coordinates below -7 , which I cured by centralising them. I centralised "*" both horizontally & vertically, and I centralised "₁₀" horizontally, slightly widening gap between the digits. The X-coordinates of the character "J" were found to range from -7 to $+8$, and whilst this could be packed into the 3 available bits, I've moved everything to the right of the low point left one unit, to agree with the Manual. In QPLOT ISSUE 1, the letter "O" and the figure "0" had a common entry in the vertex table. In the 4/11/12 version these have to have separate entries (5 extra words), and I've exploited this by making the figure "0" slightly rounder than the squarer letter "O".

I've **added** the caret character "^" which is in the Algol plotter package but was not in QPLOT ISSUE 1 (and I've made both strokes the average length of their previous unsymmetrical lengths). I've added Andrew Herbert's brackets "(" & ")" and slash "/", and I've also added brackets "[" & "]" and backslash "\". Finally I've added the relations "<" & ">" and both the acute "´" & grave "` accents. Ten internal codes remain undefined, but I've provided entries in the pointer table which point to labels (EXCLAM DITTO HASH DOLLAR PERCEN AMPERS COMMA COLON SEMICO BACK) in the vertex table, so the user can add these, or alter existing characters, by just altering the vertex table.

Regarding potential **alternate** graphical representations of the internal code, 903-Telecode's "₁₀" rather than ASCII's "?", was in QPLOT ISSUE 1, so it seemed reasonable to implement the character before "A" as 903-Telecode's grave accent, matching acute, both of which are used as string quotes in 903 Algol, rather than ASCII's "@", (which would be longer). But I've ducked the "£" issue completely. I've implemented the symbol between "[" & "]" as ASCII's "\" matching "/", not 903-Telecode's "£" (which would be longer), whilst I've labelled the alternate position HASH

with a comment that it could be a half-sign or a pound-sign.

I have placed the pointer table after the vertex table (as previously was the case in QAPLT2 but not QPLOT) to avoid a large number of forward references. This significantly reduces the size of relocatable binary tapes. I've *not* changed the pointer table into a table of half-word offsets (a technique which I have used in my legible-tape routines). This would give an overall saving, despite needing some extra code, by reducing the pointer table by 32 words and avoiding the need for some half-word filling in the vertex table. However, it would then be very difficult for the user to alter the character set.

I've completely eliminated the **error indications** from QPLOT 4/11/12. The character-drawing routine now replaces unimplemented characters by "^" (as in the Algol plotter package). Character codes outside the range 0 to 63 went undetected in ISSUE 1; now garbage in the upper 12 bits is masked out (as in the Algol plotter package). The ISSUE 1 line-drawing routine accepted parameters -1 or +1 (raise pen) -2 or +2 (lower pen) or +0 (neither), although the negative options are not in the Manual; other values gave an error message. These values still work with the 4/11/12 version, which treats +0 as requesting no change, otherwise any odd value raises the pen (always) and any even value lowers the pen (if it is up).

My 4/11/12 version of QPLOT, which includes these further changes, should still be an **exact replacement** for QPLOT ISSUE 1, except for taking less store. The character table format is no longer as described in the Manual, but the user is still free to alter the table.

As well as making these changes to the SIR routine QPLOT, I've updated the SIR sources of the **Algol package**, QAPLT1 & QAPLT2, both by incorporating the changes made to QPLOT and by improving the Algol interfacing code. I've incorporated these changes into the Algol Library Tapes for 903 Algol Issue 5 & 903 Algol Issue 6, which use different absolute addresses for communicating with the corresponding issue of the Algol interpreter. There is also an Issue 7 Library, but there was never an Issue 7 interpreter (for 8K or 16K). The Issue 7 library contains no plotter package, and it provides a slightly different implementation to Issue 6 for some of its other facilities, which are most likely to be bug fixes. So I've provided an Algol Library Tape issue 7, by adding the plotter package at the start of the existing issue 7 tape, for use with the Issue 6 interpreters (8K & 16K).

Store Saving.

QPLOT ISSUE 1 store used:

Error	61	Line	195	Char	153	Table	352	Lits	12	Total	773
-------	----	------	-----	------	-----	-------	-----	------	----	-------	-----

QPLOT 15/6/12 store used:

Error	25	Line	89	Char	96	Table	352	Lits	17	Total	579
-------	----	------	----	------	----	-------	-----	------	----	-------	-----

QPLOT 4/11/12 store used:

Error	0	Line	82	Char	95	Table	248	Lits	14	Total	439
-------	---	------	----	------	----	-------	-----	------	----	-------	-----

QPLOT 15/6/12 as a fraction of ISSUE 1:

ignoring tables $(579-352)/(773-352) = 227/421 = 54\%$.

QPLOT 4/11/12 as a fraction of ISSUE 1:

ignoring tables $(439-248)/(773-352) = 192/421 = 45\%$.

including expanded tables $439/773 = 57\%$.

The Algol line-drawing routine QAPLT1, used by SETORIGIN, MOVEPEN & DRAWLINE, and QAPLT2, which was 305 words, is now 141, saving 164.

The Algol character-drawing routine QAPLT2, used by WAY, CENCHARACTER and "print", which was 542 words, is now 379, saving 163, despite adding new symbols.

The total for the Algol plotter package, QAPLT1+QAPLT2, which was 847 words, is now 520, saving 327, despite adding new symbols.

Released Tapes.

There are several new files in the ZIP file. The names used in the ZIP were chosen to be unique in the development context and are not really relevant or appropriate in a wider context. The programs should be known by the dated names suggested for actual tapes. None of the new files have legible headers.

QPLOT_N1.SIR is QPLOT 15/6/12 in ASCII readable form.
QPLOT_N1.RAW is QPLOT 15/6/12 as a raw tape image, actual tapes should be labelled "QPLOT 15/6/12, Telecode".
This version of QPLOT retains the original table format.

QPLOT_N4.SIR is QPLOT 4/11/12 in ASCII readable form.
QPLOT_N4.RAW is QPLOT 4/11/12 as a raw tape image, actual tapes should be labelled "QPLOT 4/11/12, Telecode".
This version of QPLOT uses the more compact table format.

QAPLT1N6.SIR is QAPLT1 4/11/12 for issue 6 in ASCII readable form.
(QAPLT1N5.SIR = QAPLT1 4/11/12 for issue 5 is not included, it is the same as issue 6 except for changing "6" to "5" in the title comment and changing "BASE=100" to "BASE=0").

QAPLT2N6.SIR is QAPLT2 4/11/12 for issue 6 in ASCII readable form.
(QAPLT2N5.SIR = QAPLT2 4/11/12 for issue 5 is not included, it is the same as issue 6 except for changing "6" to "5" in the title comment and changing "BASE=100" to "BASE=0").

ALGOL3N5.RLB is the raw tape image of the Algol issue 5 library tape (tape 3) in which the plotter package has been replaced by the 4/11/12 version of QAPLT1 & QAPLT2 for issue 5, actual tapes should be labelled "ALGOL ISSUE 5 LIBRARY 4/11/12, RLB Mode 3".

ALGOL3N6.RLB is the raw tape image of the Algol issue 6 library tape (tape 3) in which the plotter package has been replaced by the 4/11/12 version of QAPLT1 & QAPLT2 for issue 6, actual tapes should be labelled "ALGOL ISSUE 6 LIBRARY 4/11/12, RLB Mode 3".

ALGOL3N7.RLB is the raw tape image of the Algol issue 7 library tape (tape 3) to which the plotter package has been **added** using the 4/11/12 version of QAPLT1 & QAPLT2 for issue 6, actual tapes should be labelled "ALGOL ISSUE 7 LIBRARY 4/11/12, RLB Mode 3".