**Line Internal format**

+0 Offset to next. If this value is 255 indicates top of RAM program, 0-2,128-254 invalid

+1 Line Number (1-254)

+2 Line as ASCIIZ string.

**Execution**

*This block is repeated, so that code automatically drops through to the end. There will be one file per one or two commands (some naturally pair up). The less speed important ones are at the top.*

Jmp EndOfCode

Execute\_Some\_Command:

*Code to execute goes here.*

EndOfCode:

*Goto, Run, If will be near the bottom because of the requirement to jump straight to execute.*

*Check error passed through (CY/L = 0, S->A, E = Error #)*

*Skip forward until either : or NULL found. If NULL and isRunning == 0 back to command line.*

*If : found step over it, if NULL, skip to next line (e.g. add 3, see above) if offset allows*

*Update line number*

*If offset is -ve then back to command line (off end of program)*

ExecuteFromHere:

*Goto, Run and If can come here as this executes without doing the step over.*

*Command line execution also comes here.*

*Eliminate : and comments, both go back to skip end.*

*Search table on first two characters, default go to LET code with no skip.*

*Code also (on success) skips n characters checking for NULL and : and then spaces.*

*Execute code with fall through to Expression evaluation via XPPC P3.*

IsRunning <byte> Non-zero if running as opposed to (say) doing a single line

Cleared on END, off end of program, ERROR etc.

LineNumber <byte> Line Number, current byte. 0 when direct.