

```

; *****
; *                               ST Secrets series                               *
; *                               by                                           *
; *                               COLM COX                                       *
; * -----                                                                    *
; *                               GRAPHICS part one - ARTICLE TWO - LISTING FOUR  *
; *                               (ASSEMBLY LANGUAGE)                            *
; *                               PAGE 6 MAGAZINE - ENGLAND                       *
; * *****                                                                    *

```

```

init           = $A000
draw_sprite   = $A00D
erase_sprite  = $A00C

```

```

;
; This code provides the interface for the majority of "C"'s available for the
; ST. This code will not work with "HABA C" - see Listing 3 for the required
; code. If you are unsure as to which code to try, try this first - it is most
; likely to work - if this code doesn't work and you are sure that Listing 6 is
; correct then try Listing 3. If this doesn't work and you are sure you've got
; no typo's - then I can be contacted at the address at the end of the article.
; But please include a S.A.E!
;

```

```

        .text                               ; Text segment.
        .global                             ; Global so that other routines in
_init_linea _init_linea                    ; link may access it.
        movem.l d0-d7/a0-a6,-(sp)          ; Save registers
        .word   init                        ; Call Line-A "INIT"
        movem.l (sp)+,d0-d7/a0-a6         ; Restore regs.
        rts                                  ; C'est tout!

```

```

        .global                             ; Global . .
_draw_sprite _draw_sprite
        link   a6,#0                        ; Push A6 and get pointer to args.
        movem.l d0-d7/a0-a6,-(sp)         ; Save regs.
        move.l 8(a6),d0                     ; X co-ordinate in D0
        move.l 10(a6),d1                    ; Y ' ' ' ' in D1
        move.l 12(a6),a0                    ; S.D.B in A0
        move.l 16(a6),a2                    ; Buffer in A2
        .word   draw_sprite                 ; Call Line-A "DRAW SPRITE"
        movem.l (sp)+,d0-d7/a0-a6         ; Restore regs.
        unlk   a6                           ; Restore A6
        rts                                  ; Bye!

```

```

        .global                             ; Global . . . .
_erase_sprite _erase_sprite
        link   a6,#0                        ; Push A6 and get pointer to arg.
        movem.l d0-d7/a0-a6,-(sp)         ; Save Regs.
        move.l 8(a6),a2                     ; Get buffer in A2
        .word   erase_sprite                ; Call Line-A "ERASE SPRITE"
        movem.l (sp)+,d0-d7/a0-a6         ; Restore regs.
        unlk   a6                           ; Restore A6
        rts                                  ; THE END.

```