
 Programs

```

Line 100 initializes lower 8k | >100 CALL INIT
Line 110 loads the assembly | >110 CALL LOAD(9838,47,0,38,1
program shown below. VMBR | 14,4,32,32,44,3,128)
Line 120 loads registers with | >120 CALL LOAD(12032,0,0,48,0
VDP address, Buffer, Length. | ,2,255)
Line 130 runs line 110 program | >130 CALL EXECUTE(9838)
Line 140 loads the assembly | >140 CALL LOAD(9838,47,0,38,1
program shown below. VMBW | 14,4,32,32,36,3,128)
Line 150 loads registers with | >150 CALL LOAD(12032,0,0,48,0
VDP address, Buffer, Length. | ,2,255)
Line 160 runs line 140 program | >160 CALL EXECUTE(9838)
Line 170 put a command in here | >170 CALL VCHAR(1,1,32,768)
Line 180 loops to line 160 | >180 GOTO 160

```

| HEX ADDRESS | HEX VALUE | ASSEMBLY COMMAND EQUIVALENT |
|-------------|-----------|--------------------------------------|
| >266E | >2F00 | DATA >2F00 (workspace area address) |
| >2670 | >2672 | DATA >2672 (start execution address) |
| >2672 | >0420 | BLWP (first executed command) |
| >2674 | >202C | @VMBR (or >2024 VMBW) |
| >2676 | >0380 | RTWP |

| | | |
|-------|-------|---------------------------------|
| >2F00 | >0000 | REGISTER 0 (VDP address) |
| >2F02 | >3000 | REGISTER 1 (RAM buffer address) |
| >2F04 | >02FF | REGISTER 2 (length of text) |

Normal XB using LINK.

```

Initialize for Assembly. | >100 CALL INIT
Load support routine. | >110 CALL LOAD("DSK1.TEST")
LINK to program. | >120 CALL LINK("GO")
RXB EXECUTE EXAMPLE. |
Initialize for Assembly. | >100 CALL INIT
Load support routine. | >110 CALL LOAD("DSK1.TEST")
EXECUTE program address. | >120 CALL EXECUTE(13842)

```

EXECUTE does no checking so the address must be correct. The LINK method finds the name and uses the 2 byte address after the name to run the Assembly. EXECUTE just runs the address without looking for a name thus faster.

Options.

Dependent on Programmers use and skill.