# Patching HEPAX

**The DISASM function in the HEPAX module switches off the 41CL immediately after the final address digit is entered for the command.**

The DISASM function scans the keyboard during the disassembly process, but it appears that to save space this scanning function was not implemented properly. (The code does not look at the keyboard valid flag.)

In addition, it appears that the 41CL keyboard scanner does not output the same code as the original 41C when no key is being pressed. (The idle state code was not specified in the HP documentation.)

As a result of these two issues the HEPAX DISASM code thinks that the ON key has been pressed immediately after the last address digit has been entered, turning the calculator off.

The way around this issue is to remove the test for a press of the ON key during the DISASM function. This requires copying one page of the HEPAX code to RAM so that one location can be patched, and then pointing the MMU at the patched code.

The example below assumes that the HEPAX module has been loaded into the lower half of Port 2, which is page A, and that the uppermost page of RAM (starting address 0x83F000) will be used for the patched HEPAX page.

First, the Bank 4 HEPAX image is copied to RAM:

```
ALPHA 030>83F ALPHA
XEQ ALPHA YMCPY ALPHA
```

Next, the instruction that tests for a press of the ON key is replaced with a NOP instruction:

```
ALPHA 83F08D-0000 ALPHA
XEQ ALPHA YPOKE ALPHA
```

Finally, this RAM page is substituted for bank 4 of the HEPAX image in Flash by directly programming the MMU register. The MMU register must be programmed directly because we are only substituting one bank of the HEPAX code.

```
ALPHA 8040AC-883F ALPHA
XEQ ALPHA YPOKE ALPHA
```

**The HEPAX DISASM function does not allow the disassembly of the HEPAX code itself.**

If you want to remove this restriction, four locations in Bank 1 of the HEPAX code need to be modified. The code is in Bank 1 of the HEPAX code, and I will use the RAM at address 0x83E000 to hold the patched code.

First, the Bank 1 HEPAX image is copied to RAM:

```
ALPHA 02D>83E ALPHA
XEQ ALPHA YMCPY ALPHA
```

Next, the instructions that branch to an error routine are replaced with NOP instructions:

```
ALPHA 83E131-0000 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 83E132-0000 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 83E133-0000 ALPHA
XEQ ALPHA YPOKE ALPHA

ALPHA 83E134-0000 ALPHA
XEQ ALPHA YPOKE ALPHA
```

Finally, this RAM page is substituted for bank 1 of the HEPAX image in Flash by directly programming the MMU register:

```
ALPHA 8040A0-883E ALPHA
XEQ ALPHA YPOKE ALPHA
```

**No exhaustive testing of these patches has been done. Once enough users have verified that there are no unintended consequences, the patched versions can be written back to Flash memory.**