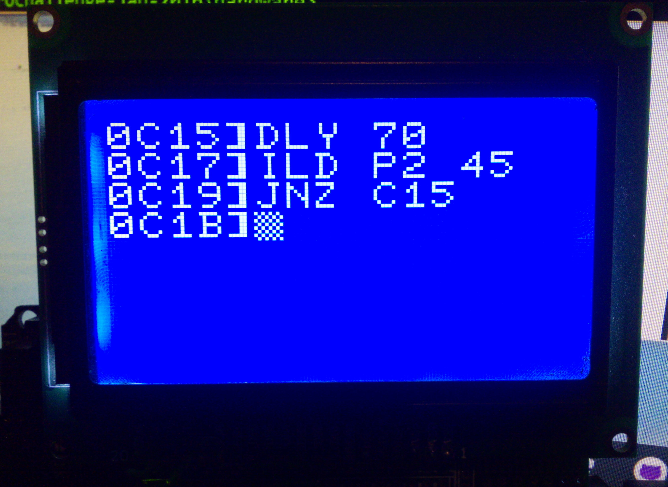
**Release : Post14 Software in software/monitor\_rom**

Well, I now have a working inline assembler, which was fairly easy – a lot of the work had been done in the rest of the ROM.

I still have plenty of space. I was worried that what I’ve done to date – memory read/write/view/execute, tape i/o and the assembler would take up the whole ROM, as of the time of writing the space between 0x3CC and 0x751 is empty, which is 901 of the 2,048 bytes I have available.

A quick demo showing it running on the replica hardware. I have written three lines of SC/MP assembler here which are a slightly modified version of full assembler. In a proper assembler it would look like

Loop: DLY 70

ILD 0x45(P2)

JNZ Loop

So you can see the correlation quite easily.

(DLY is software delay, the operand says how long, ILD is increment and load memory at address P2+$45 and JNZ is jump if not zero)

In the second picture I have typed C (to clear screen) and D C15 (Dump memory from C15 onwards) and you can see the results.

Note that the relative address calculation on the JNZ has been calculated automagically from the C15 (the absolute target) to $FA (the 8 bit signed relative target).

It isn’t super intelligent, it doesn’t (for example) check the range on relative offsets, so if the actual offset is $105 not $005 it will still assemble it. But it’s only really intended for very simple programming.

The next thing to do is the labels (so you can do JNZ =7 rather than remembering addresses) and to work a disassembler. Then I will see how much space is left, and probably fill the rest with utility routines, maybe.