

SP80A

1. Remove all leads, add-ons etc from your Spectrum and turn it over to expose the 5 cross-headed bolts.
2. Remove these 5 bolts.
3. Holding the case together with your hands, turn the Spectrum right-side-up. The top of the Spectrum case is now attached to the rest by two rather fragile ribbon cables.
4. Carefully lift the top of the case towards you taking care not to strain the two ribbon cables so that the two vacant sockets are easily accessible on the Spectrum circuit board.
5. Remove the polystyrene protection blocks from the pins on the underside of the SP80 board.
6. Carefully plug the SP80 into place over the vacant sockets making sure that the 14 pins enter the 14-pin socket and the 16 pins enter the 16-pin socket (the 2 red LEDs are situated above the expansion port on the Spectrum).
7. Power on the Spectrum. Wait for the standard message to appear on the screen. you should have to wait a bit longer than you had to with just 16K of RAM on the system. Both of the red LEDs should have lit up.
8. Type `OUT 65407,1` this will switch in the extra 32K block of memory (and cause the Spectrum to re-initialise first round time). One of the LEDs should have gone out.
9. Type `OUT 65407,0` this will switch out both 32K blocks leaving only 16K for Basic but allowing protected retention of contents of both 32K blocks until they are switched in again. Both lights should go out.
- 10 Type `OUT 65407, 3` this will switch in the 32K block which comes on automatically when the Spectrum is turned on. Both LEDs should come on.
11. The 2 LEDs may be left extruding from the Spectrum expansion port or may be bent up over the SP80 board inside the case as desired.
12. Reposition top half of case and replace screws.

USE. - There are a number of ways of using the extra 32K of memory eg.

1. Storage of raw data using PEEK and POKE.
2. Storage of several screens.
3. Machine code routines.
4. Additional Basic arrays.

A booklet containing software and examples of these and other uses will be available from us in October price £2-00 + 20p P&P.

To switch the memory blocks from a machine code routine it is necessary to load the A register with 255 before executing an `OUT(127),x` (where $x=0,1,2$ or 3).

To prevent either of the 32K blocks from coming into use when turning the Spectrum on it is simply necessary to remove the 68ohm resistor (blue, grey, black, gold) from the SP80 board. They can then be switched on by software at a later stage.

EAST LONDON ROBOTICS
Finlandia House
14 Darwell Close
East Ham
LONDON E6 4BT

Tel.01 471 3308

SP80 Technical Data.

Compoents: 8 X Industry standard 64K bit Dynamic RAM (Currently NEC 6164 150ns)
5 X Low power Schottky packages LS08,LS27,LS75.2xLS157/8.
Several discrete components.

Board: Fibre-Glass with solder resist. Overall dimensions less than 1½" X 6½"

Dynamic RAM controller circuitry: All control and address lines buffered. CAS decode.

Power consumption: 200ma nom. Characteristics virtually identical to Sinclair 32K.

Paging Control: A7 low on OUT instruction selects 2 bit control word on D0 and D1 thus –

OUT 65407,1	brings in additional 32k of memory.
OUT 65407,3	brings back original 32k of memory.
OUT 65407,0 or 2	Takes out both 32K blocks of memory but without letting them lose their contents.

The same effect can just as easily be accomplished directly from a machine code routine using Z80 OUT instruction (fully described in operating instructions).

NOTE – On power on the normal 32K block is automatically switched on to make things exactly like a 48K Spectrum. This may be modified if so desired by disconnecting of a single component from the SP80 board.

EAST LONDON ROBOTICS – THE OPTIMUM SOLUTION

EAST LONDON ROBOTICS
Finlandia House
14 Darwell Close
East Ham
LONDON E6 4BT
Tel 01 471 3308

ZX 16K RAM Pack - Instructions for use

This RAM Pack will expand the memory capacity of your Sinclair Personal Computer to 16K bytes (1 K byte - 1024 units of storage, i.e. 1024 characters) - which is the maximum it can handle. The ZX 16K RAM is not expandable in stages and so is supplied with the 16K of memory built in. All you have to do is plug it onto the expansion edge connector at the back of your computer. It will take its power from the existing power supply.

IMPORTANT — Make sure that the computer is switched off before plugging or unplugging the memory. Otherwise you may damage the RAM pack or computer — also the computer's operation will be interrupted for a short while and it may get stuck in a 'non communicative' state. You will then need to switch off and on again anyway. Any program that was in the computer will certainly be lost; so if you think a particular program may need a lot of memory, it is wise to plug on the RAM pack before entering it. As a guide, the computer's internal memory can handle 30-60 lines of program, but things like large arrays or a lot of data can cut this down considerably.

A little mental arithmetic should show that the computer can theoretically handle up to 960 program lines when fitted with the expansion pack. In practice, no program is likely to need this much (No prizes for trying!; except programs" with a lot of data. In fact, the only evidence the user sees of the RAM pack's operation - apart from almost never running out of memory - is that the inverse 'K' takes a little longer to appear after switching on. This is because the computer tests the memory after switch on to see how much there is. The results of the test aren't directly available, but a simple machine code program can be used to reassure the operator that he or she really has got the memory available. :

MEMORY TEST PROGRAM

```
1 POKE 18000,33
2 POKE 18001,11
3 POKE 18002,0
4 POKE 18003,57
5 POKE 18004,68
6 POKE 18005,77
7 POKE 18006,201
8 PRINT (USR(18000)- 16373)/1024; "K"
```

If run on your computer without the RAM pack the answer will be I K - the amount of memory present in the system. With the 16K RAM pack, however, the answer will be 16K. Note that this program does not say how many bytes are free for a program, only how many are available to the computer, which will use some of them for the display file stack, etc.