

BBC RAM/FLASH board assembly

Parts you will need.

The PCB

1 x 74LS00

1 x 74LS139

1 x W24512AK-15 or similar

1 x AM29F010

5 x 10n caps

40 pin length of turned pin socket

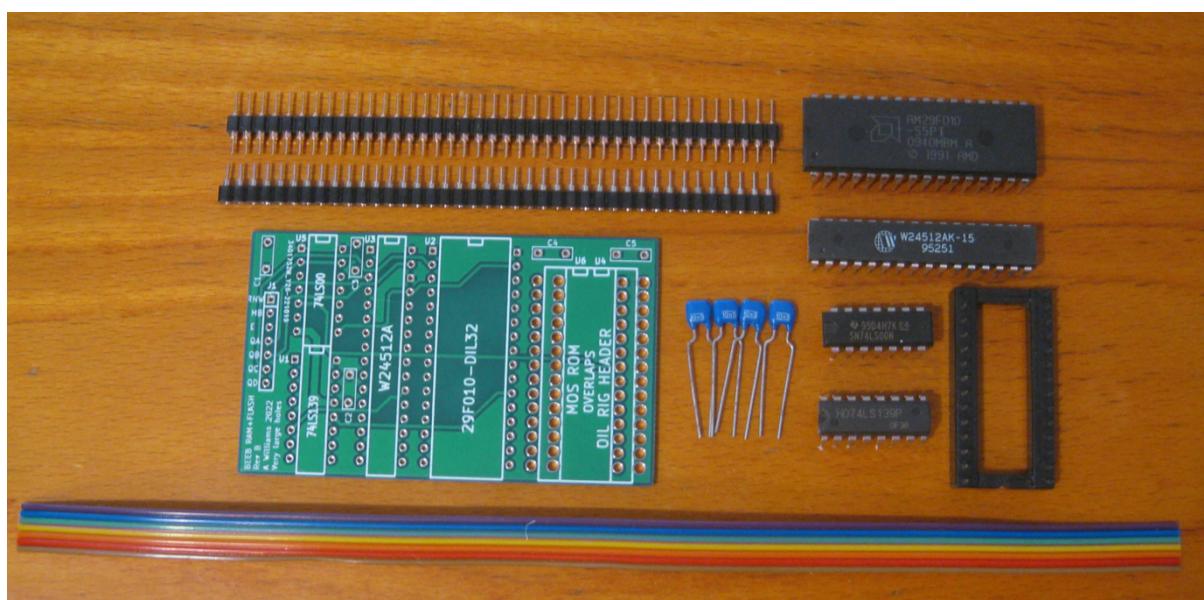
40 pin length of turned pin strip

1 x 28 pin chip socket not turned pin

length of rainbow ribbon cable 7 cores, about 120mm (or anything you have)

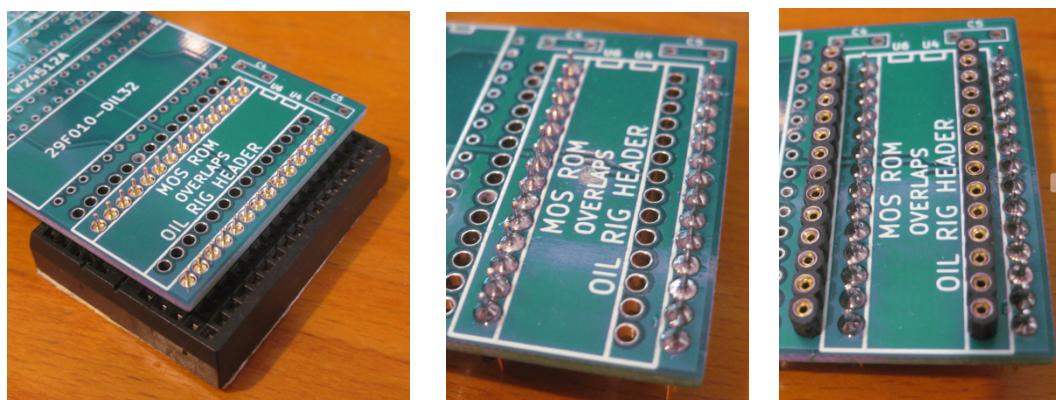
Optionally 5 chip clips and 2 female dupont lead ends.

At the time of writing all of the parts other than the PCB were available from Aliexpress if nowhere nearer.



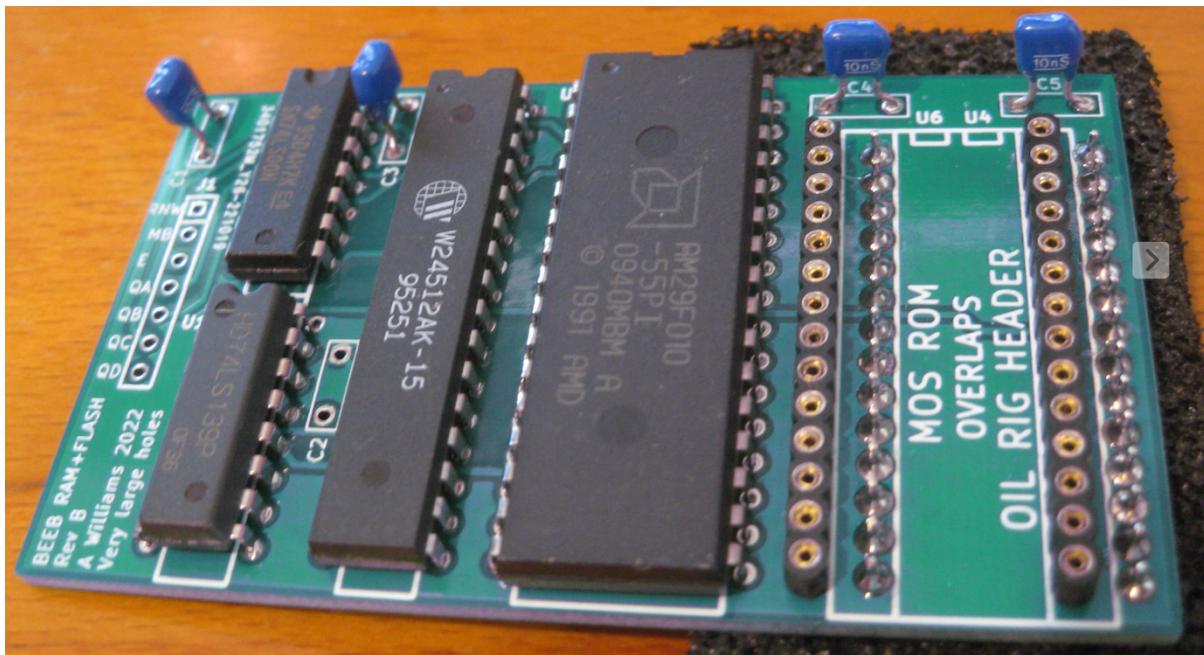
(note you will need five capacitors not four as shown above)

The large holes on this board allow the turned pin connectors to sink deep into the board so it fits low enough to not foul the keyboard. I use the scrap solderless breadboard to keep everything vertical while I solder the pins in.



You could use a chip socket in place of the socket strip but the cross members foul the upward pointing ends from the oil rig so some trimming of both would be necessary.

With the connectors fitted solder the rest of the components in. They are all different sizes so its hard to get it wrong. Solder the bottom right then top left pins of the chips first as this improves their ESD tolerance stick the oil rig into some ESD foam too.



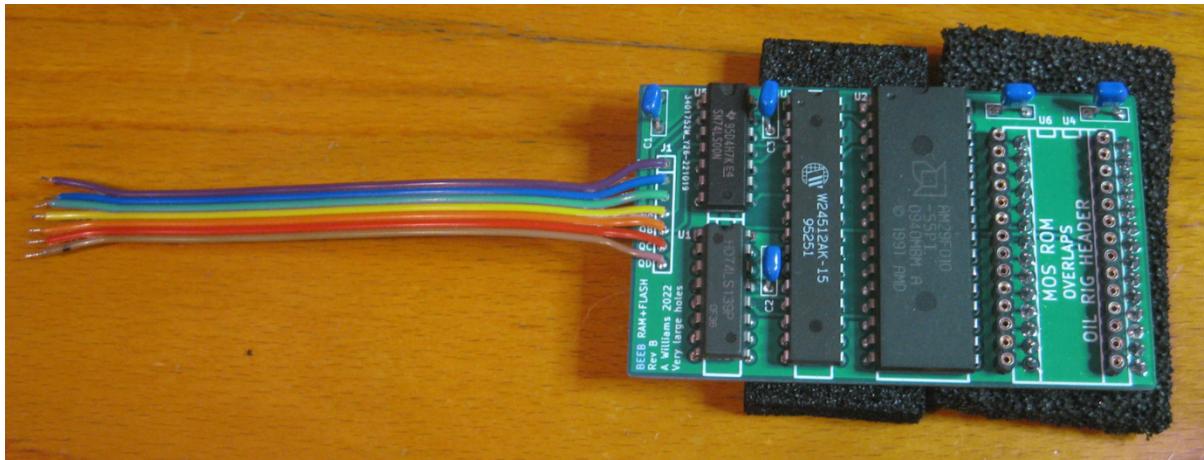
(I hadn't noticed I had still missed a capacitor even by this point)

Splay the ends of some rainbow ribbon back to about 14/15mm



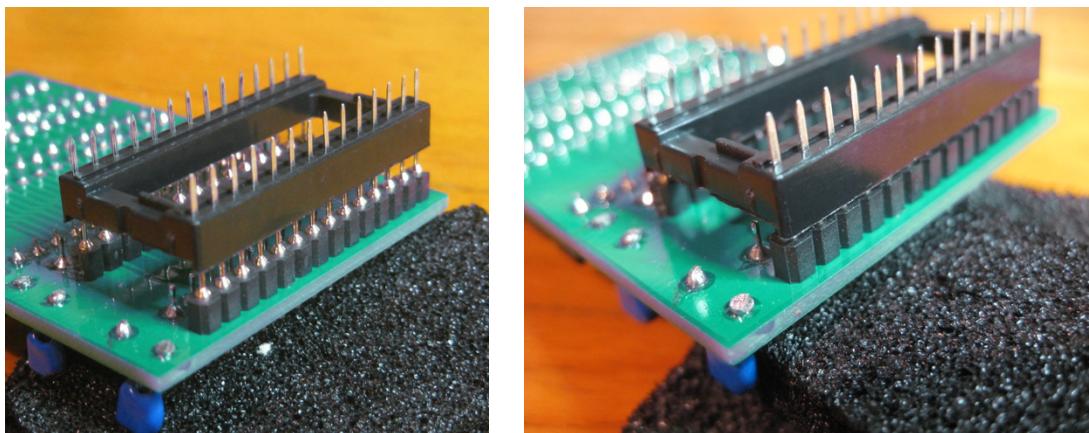
Cut the ends straight across with some scissors (Don't get caught doing that or you will be in a whole world of pain with the scissors owner) Strip back about 4mm tin and solder into the PCB





The ordinary chip socket is going on the turned pins to finish the oil rig.

I use the solderless breadboard again to squeeze them together.



Using the ordinary chip socket in this way saves damage to the motherboard sockets. They don't stand up to the tuned pins for many insertion/extractions.

Fitting.

If you don't want to solder to your beeb you can use chip clips and dupont connectors which you will have to solder to the ends of the leads first.



First remove the MOS rom chip from the mother board (left most ROM IC 51) and relocate that into the FLASH RAM board with the same orientation notch towards the capacitors.



If you think your motherboard socket might be a bit worse for wear already then carefully twist each pin of the oil rig socket by about 45 degrees. You can do this with ROMS or other chips too. It brought my Filestore back to life.

Its easier to solder wires yellow to brown to IC 76 the ROMSEL latch if you do this before plugging the board in to the MOS socket.

16

15

14 – QA yellow

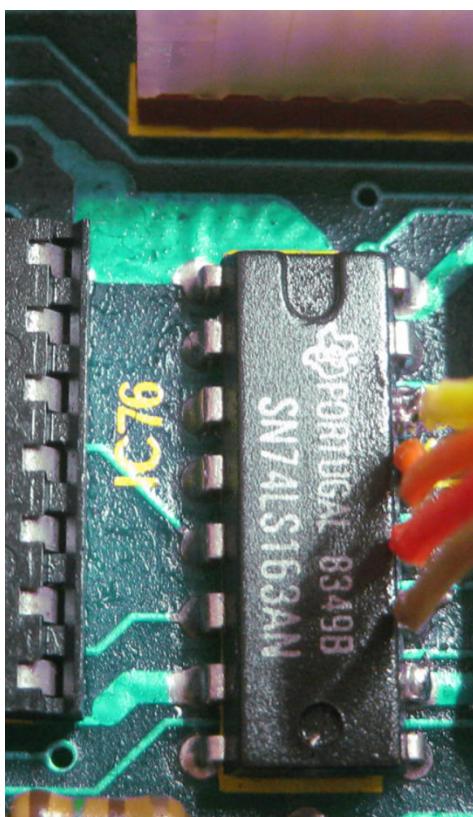
13 – QB orange

12 – QC red

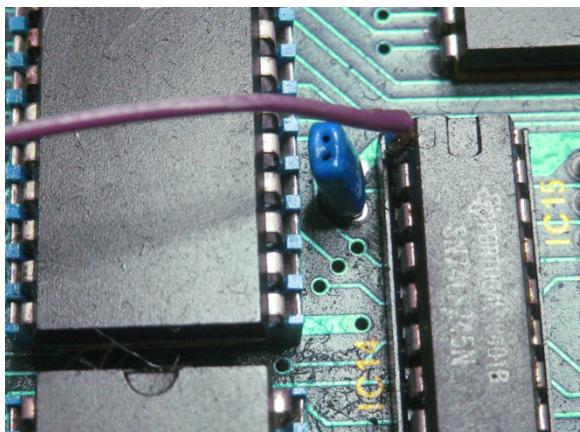
11 – QD brown

10

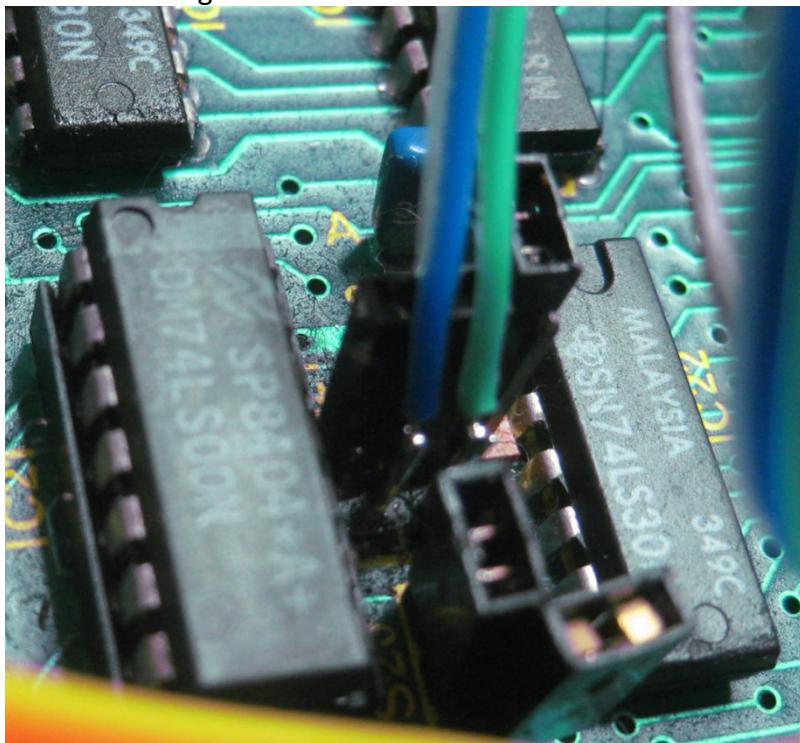
9



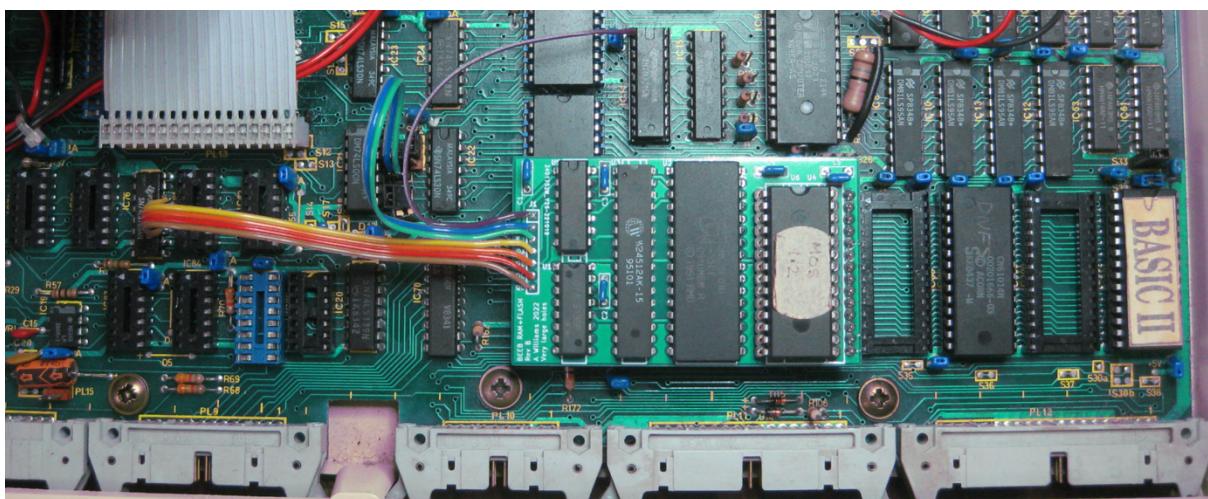
Then purple to pin 1 of IC 14



Then blue and green to S21.



Now insert the oil rig into the motherboard MOS socket.



The machine should power up normally now.

The rest of the repository contains the kicad 6 design files. These are the files from which this board was made by JLCPCB. I have not converted them to a later version since then they

would not be trusted or tested.

There is a BBC ssd file with the software and also a copy of all the source as my RISC PC saves them to LanManger. I build the software on a RISC PC with 65tube. The software is there in tokenised basic, listed basic in text and 6502 binary forms.

Much of this code dates back to the late 80s.

The commands are pretty self-explanatory. *SRLOAD is the one you will want the most and it tries to follow the M128 syntax except its always in 'Q' mode. (Use main RAM to buffer the ROM in one go wiping out anything that was there before)

As far as the sideways RAM is concerned these utilities should be interchangeable with other implementations. Its likely you will only be able to use these implementations with the FLASH ROM since writing them is not as simple as treating them as RAM.

The header generator should be useful with other sideways RAM implementations. Its job is to save a rom out of the machine with a loader tacked on the front. The aim is to make the user of the ROM transparent as to its being in the machine or not.

For example, if you use it to create a file 'BE' from the BASIC Editor and you copy that file to a library directory then *BE will work normally if its in the machine and will load it to RAM or FLASH and start it as the current language if its not.

This was written originally for a school where some machines would have the roms and some would not. With headered roms in the library the instructions to the kids stayed the same either way.

The 80's version of this board had ROM and SRAM, but it could write protect the SRAM. Some of the legacy code for that functionality may still be in the source but its not supported by this hardware because you can use the FLASH instead.

Thanks go to a post that I can no longer find on an Apple2 forum which explained the issues with trying to program this 29F010 on a 6502, this explained that it had to be done with self modifying code to avoid the 6502's dead read cycles breaking the write contract with the chip.

Also to the star dot forum members who have said they want to build one, please upload the files.

Enjoy.

Alan