

BATTERY BACKUP RAM FOR THE n8vem

The N8VEM is a single board CPM computer with 512k of RAMDISK.

http://www.instructables.com/id/Robot_Brain_Build_a_single_board_computer_in_an_e/ In its native form the RAMDISK is wiped on powerup. Further, it is not static unless one buys a more expensive battery backed RAM module. These modules cost three times the price. This project uses a standard 512k static ram chip and a DS1210 chip to protect the memory on powerdown. The backup can be provided either by a supercapacitor (days of storage), or two AA alkaline batteries.

PARTS

The N8VEM comes as standard with a 1meg EPROM and a 1 meg ram. I used a 628512 512Kx8 CMOS RAM from Futurlec and picked up a few DS1210 chips for \$2.75 while I was there. Futurlec have fantastic prices but they do take 10-14 days to deliver, so plan ahead. Other parts were some 28 pin sockets, a 28 pin ZIF socket, 470uF cap, 5V reg with heatsink, green led, 220R resistor and a 0.047 farad super capacitor. Have a read through the text first, because you may not need some of these components. I could have used a couple of AA batteries in a battery holder instead of the supercap, but I wanted to double check the current draw really was as low as the claimed 10 microwatts (it was).

THE BOARDS

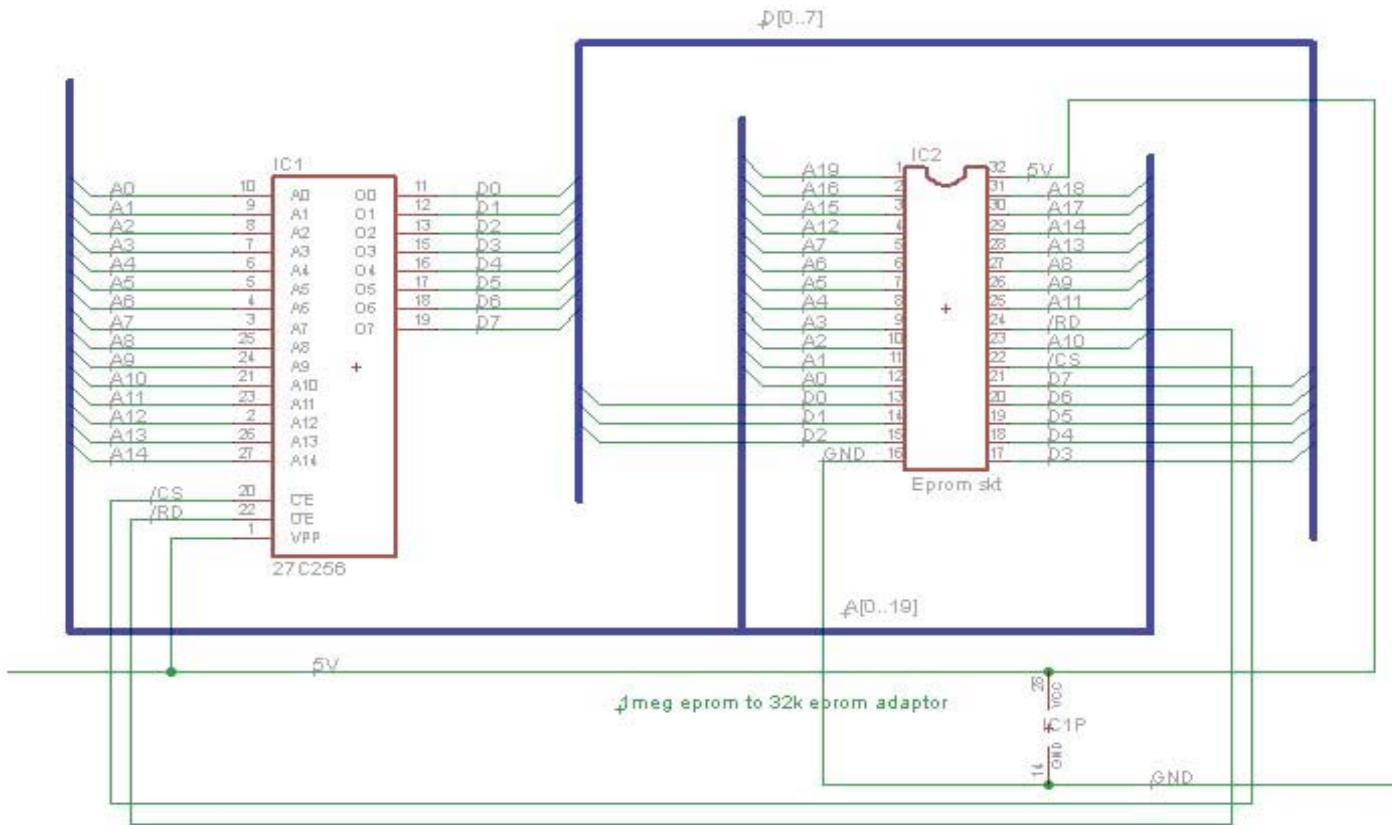
This project involves some hacking, so to play it safe and to double check each change didn't upset something else, I had two boards next to each other – a working one and the one being modified.



The modified board is the one at the top. First step was to modify the EPROM, and this step isn't really necessary if you just want to grab the code and use it as-is on your existing 1meg eprom. However, I'll describe this step as it could be useful if you want to modify files. The N8VEM comes with a 1meg eprom and these are expensive. This project took over 100 reprogramming cycles, and I had lots of 27C256 32k

eproms handy but only a few 1 meg ones. Others have suggested using an EEPROM and this would work fine too. But I had the eproms and I have an eprom programmer so it was easier to just use them and erase them in batches.

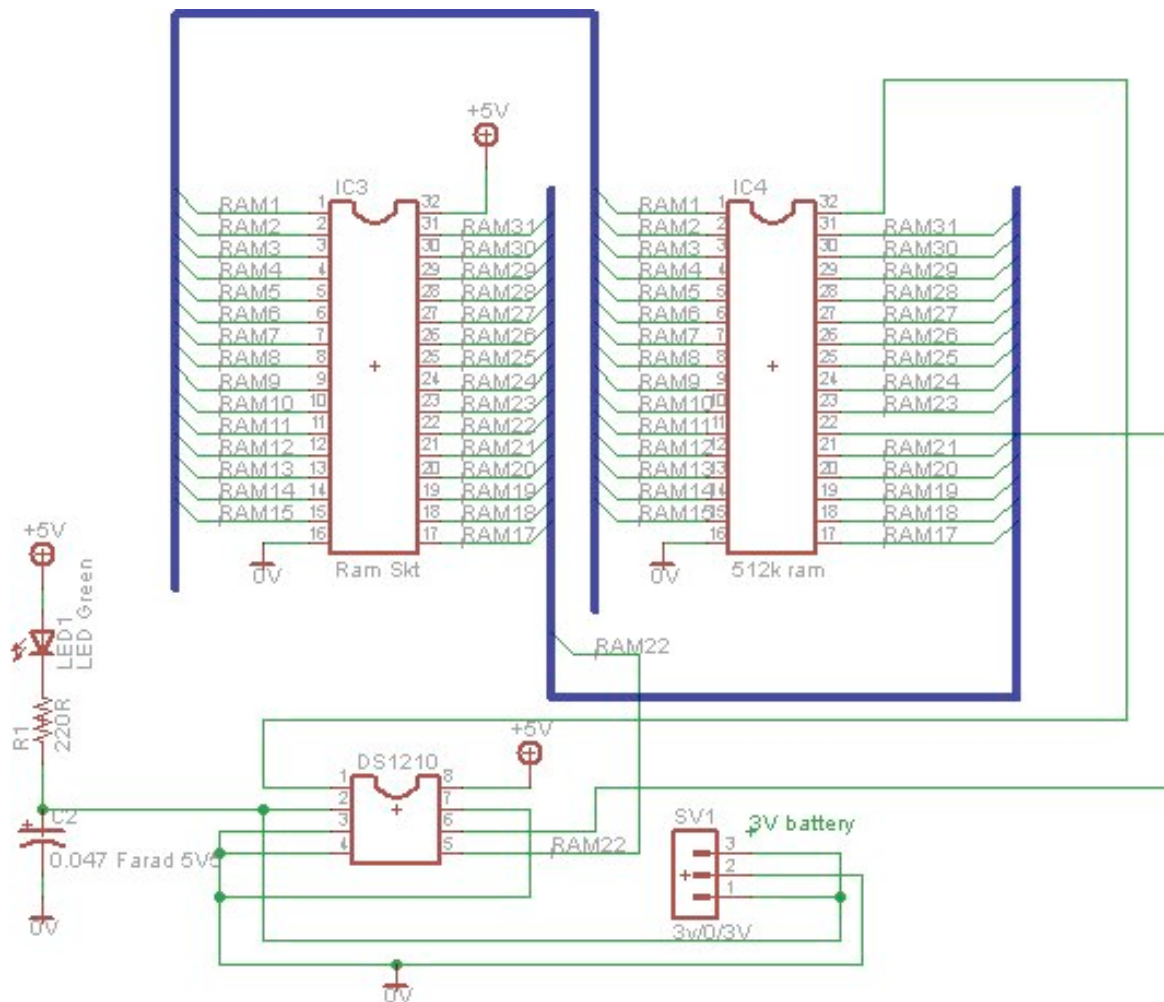
Modifying the eprom involved building a little tower out of 28 pin sockets, so the ZIF socket was high enough to clear the other chips. On one layer of this “tower of socket”, a few minor mods were made. Only a couple of wire links were needed as the 32k and 1meg eproms have almost the same pinout.



This is the formal schematic. It really is just some mods at the pin 1 end of the eprom.

The final result is a ZIF socket as this can handle the multiple insertions.

Next step is to wire in the DS1210. This chip sits between the power supply pin of the RAM chip (pin 32) and also sits between the CE pin 22. The power supply pin was removed from the socket and a wire soldered onto it (You could do some clever mods to the pin 32 on both sides of the board if you haven't soldered the board up yet).



And not shown is a 470uF on the board between 5V and 0V. This may or may not be needed but it should help slow down the collapse of the supply voltage which will help the DS1210 switch over cleanly to the backup supply. I'd probably use a couple of batteries rather than the supercap as the batteries will last for years.

There is one other modifications to the board that is most important, and that is to increase the value of the reset cap C2 from 0.1uF to 100uF. This greatly slows down the Z80 starting up to about half a second. However, and this is the good news, the delay on startup was mostly due to reformatting the ram disk, and because this is only ever done once now, the overall startup delay of the board is now significantly **faster** – ie about half a second!

SOLDERING UP THE BOARD.

Ok, this was a test jig and when it was soldered up it was not clear it was going to even work. So apologies for “dead bug” wiring. Ultimately this might entail adding the mods to the next revision of the N8VEM board but for the moment, this is about half an hour’s work of wiring.



CPM SCREENSHOT

Test Prototype Monitor 32K-1M EPROM & Permanent RAM

CP/M drives are A, B and F

T to format B drive for first use (erases files)

Type A for help, C for CP/M

>A

D XXXXH YYYYH Dump memory from XXXX to YYYY

E XXXXH Examine memory location XXXX

F XXXXH YYYYH ZZH Fill memory from XXXX to YYYY with ZZ

H Load intel hex format data

I Input from port and show hex data

K echo keyboard input

M XXXXH YYYYH ZZZZH move memory block XXXX to YYYY to ZZZZ

O Output to port hex data

P Program ram starting from current location

R Run a program from current location

T Format RAM disk B (format before first use)

'ESCAPE' Stop current mode and wait for new command

C Boot CP/M from ROM drive

>C

Loading CPM...

CP/M-80 Version 2.2C for the N8VEM, October 2008

Run XM from A drive, this downloads file to B drive

```
A>B:
B>DIR
B: MYFILE  TXT : MBASIC  COM : HELLO  BAS
B>A:
A>DIR
A: GM      COM : RTC     COM : XM     COM
```

CODE MODS

Now to grab some code. To make it easy, just get ROMIMAGE.BIN. But if you want to modify the code further, have a read further down.

In the end the modifications were as follows:

CPM2.2 – changed the signon message so it has a reminder about how to use XM

XM.COM – this is a working copy and I have no idea which .asm it came from (but it wasn't the latest one on the N8VEM website as the latest one doesn't like my loopback cable). So this is a .COM file and the batch assembler file does not compile this one but just includes it.

Loader-m.asm – this has been extensively modified with easier to understand signon messages.

Dbgmon-m.asm – this started with some changes but in the end went back to the original and only the name has been changed.

Build-m.bat – this is the batch file needed to compile all the new files.

ROMIMAGE.BIN – if you can't be bothered making any mods.

These files are all in the attached zip file.