

SHARP PC-1600 “superRAM” Memory Expansion

256KB / 2x256KB / 512KB RAM Custom Module – Revision 2



IMPORTANT:

- Before kit assembly or use, please read these instructions.
- Avoid touching the contacts of an assembled module (risk of damage by static electricity).
- Insert and remove the module only when the computer is switched off.
- The module is designed for use in slot 2 of the SHARP PC-1600 and is not suitable for any other types of computers, even if it fits mechanically - such as in the SHARP PC-1500 (A).

1 Introduction

The *superRAM* memory expansion module supports three operation modes with different advantages (and trade-offs), so you can choose the appropriate mode with respect to your requirements:

- **256 KB Mode (default)**
Advantage: Largest possible native mode. No PEEKs and POKEs or additional driver- or init-software is required, the PC-1600 recognizes the *superRAM* module automatically in this mode. No additional hardware switch needed.
Trade-off: Only half of the modules memory capacity is used.
- **2x256 KB Mode**
Advantage: Full usage of the modules memory capacity with no additional driver- or init-software needed.
Trade-off: The memory capacity is partitioned into two separate banks, 256KB each. You need an additional hardware toggle to switch between those banks. Soldering required.
- **512 KB Mode**
Advantage: Full and unpartitioned usage of the modules memory capacity. No additional hardware switch needed.
Trade-off: A custom init-routine is required. So you need an upload option for binary files from your host computer to the PC-1600 in order to get this mode running.

As far as I know this is the first RAM module for the SHARP PC-1600 that provides a 2x256KB and even a 512KB mode. The largest original RAM module from SHARP is the CE-1601M with 64KB.

By default the module is not battery buffered, that is, the data stored on it is retained as long as the module is mounted in the slot, and the PC-1600 is connected to a power source (batteries or AC adapter). The mounted module will retain its data when the computer is switched off (and a power source is connected). You can “bridge” a battery change by temporarily using an AC adaptor without losing data. Additionally a battery buffer modification can be applied (see section 10).

2 Hardware Installation

Operation in slot 1 is technically possible, but makes little sense since it would there behave like a 32KB RAM module.

Switch off the PC-1600 and install the module in slot 2 as shown in the picture. Components must be visible.

Next you need to configure the desired operation mode and then initialize the module. The required steps depend on the selected mode and are described in the corresponding section below.

After successful initialization, seal the slot with its original cover, the module easily fits.



3 Hardware Configuration

There are three valid states or bridges – called *Options* – which are shown in the following picture. I refer to the term *Option* regardless of its implementation as a jumper, a fixed wire bridge or with aid of an additional toggle switch:



CAUTION:

- Don't use other or more bridges at a time, otherwise the module and/or the PC-1600 may be destroyed!
- One of these Options must be selected when the PC-1600 is switched ON, otherwise the module will be unstable and have increased power consumption.

Now, here is the relation between these Options and the operation modes, introduced above:

- **256 KB Mode =>**
Statically select **Option 1**
- **2x256 KB Mode =>**
Provide an additional hardware switch that toggles between **Option 1** and **Option 2** (See section 5 for details)
- **512 KB Mode =>**
Two phase initialization with **Option 1** (temporary) and **Option 3** (final) involved.

4 256KB Mode

The initialization and use of the 256KB mode in slot 2 is as with the original module CE-1601M (64KB). The total amount of memory can be configured as a RAM disk (S2:) or 32KB of it as an expansion of main memory (S0:). Here are the concrete configurations:

Basic configuration: 256KB RAM disk (S2 :), no main memory expansion

S0 configuration: 32KB main memory expansion (S0:) + 224KB RAM disk (S2:)

The corresponding initialization procedures are as follows:

4.1 Basic configuration

- Switch off the PC-1600 and install the module (see section 2)
- Select **Option 1** as described above
- Turn on the computer - `NEW0?` appears on the display
- Inputs:
CL-button
NEW0 (in PRO mode)
TITLE
- Initialization as RAM disk:
Input: `INIT"S2: ", "F"`
- Check the RAM disk capacity:
Input: `DSKF "S2: "`
Output: 251904
This is the effective usable capacity.

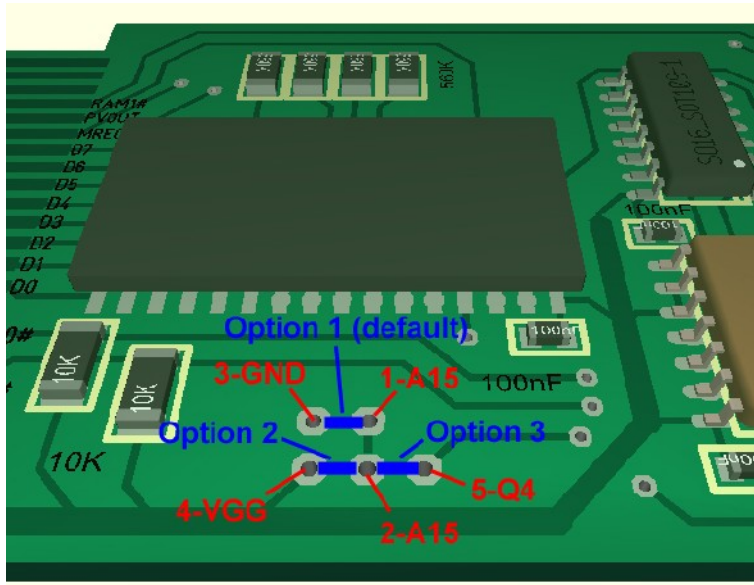
4.2 S0 configuration

- All steps of the basic configuration (see above), then:
- Initialize 32KB main memory expansion (for BASIC or machine language):
Input: `INIT"S2: ", "M"`
- Check the main memory capacity:
Input: `MEM`
Output (depending on the module in slot 1):
44xxx (no module in S1 :) to 77xxx (32KB module in S1)
- Check the new RAM disk capacity:
Input: `DSKF"S2: "`
Output: 220160

Note: Instead of `INIT"S2: ", "M"` also `INIT"S2: ", "P"` may be used in order to define a separate Program Module". For details, see the PC-1600 manual.

5 2x256KB Mode

By providing a hardware switch between the two 256KB banks (Option 1 and Option 2), you can realize a 2x256KB module.



Since the pads/pins 1-A15 and 2-A15 are connected by PCB-design, you may choose one of them as fixed input for the toggle switch. The switch then has to connect the A15 signal to either 3-GND (i.e. 256KB bank 1) or 4-VGG (i.e. 256KB bank 2) and toggle between them.

The initialization procedure is identical to that of the 256KB Mode, except you have to perform it for both 256KB banks separately. I recommend using the same configuration for both banks.

Be careful: An unreflected usage of the S0-RAM-Extension (`INIT"S2: ", "M"`) in this mode may have side effects like programs ripped into parts when switching between the 256KB banks. But in principle its working and you do not need additional initialization software, like with the 512KB mode.

6 512KB Mode

As stated above, the `INIT`-command of the SHARP PC-1600 does not support a 512K RAM expansion (but the hardware does). So you will get an `ERROR`, when you try to `INIT` or access the S2: RAM disk in 512KB mode.

But the good news is, that in fact it is possible to dupe the operating system with a custom patch routine. I have written such a program (`SUPERRAM.BIN`), which I can provide separately. You'll need an upload option from your PC/MAC to the SHARP PC-1600, that is capable of binary transfer mode. The program detects whether your `INIT`-ialized S2 RAM disk has a size of 256KB or 224KB and patches the capacity and header data of the RAM disk to 512KB or 480KB respectively.

A fully functioning 512KB RAM module for the PC-1600 might be a “world premiere” :)



Since `SUPERRAM.BIN` is not a full replacement of the `INIT`-command, but a patch routine only, you need to toggle between Option 1 (256KB) and Option 3 (512KB) in order to get the 512KB initialization done. After initialization you must not switch back from Option 3 (512KB Mode) unless you want a re-initialization. Since you have to switch only once per initialization an additional hardware toggle switch is not necessary – the jumper is sufficient.

Here is the 512KB Mode initialization procedure:

1. Set (the jumper to) Option 1. **Option 2 or Option 3 would NOT work!**
2. Initialize the module as documented in section 4 with "F", "M" or "P" parameter as you wish.
So `DSKF"S2 :` should show 251904 or 220160 respectively.
If you have configured an S0 main memory expansion of 32KB, it will be preserved by the patch routine resulting in a 480KB RAM disk at S2.
3. Load the provided program SUPERRAM.BIN (see below)
4. `CALL &C0C5`
If everything is correct, you get a message that the S2:RAM disk has been patched
5. Switch off the computer
6. Set (the jumper to) Option 3 (512KB Mode)
7. Switch the computer on.
8. `DSKF"S2 :` should show 514048 or 481280 respectively – done.

Note:

The SUPERRAM code does not (and cannot) check, whether the module inserted in slot 2 has a potential hardware capacity of 512KB. It only checks for a 256KB resp. 224KB RAM disk and then patches its header. So **do not forget step 6**, otherwise your data stored on the RAM disk will be corrupted.

There are three supported methods for the custom initialization program upload (step 3):

6.1 Upload via CE-133T / CE-140T

The standard hardware for file transfer to/from the SHARP PC-1600 is the CE-133T or CE-140T (utilizing a USB2Serial adapter) via RS232. The CE-133/140T is basically a RS232 voltage level converter. A terminal program like hterm (Windows) or CoolTerm (MAC) is used then in binary transfer mode. If your connection (basically the USB2Serial adapter) is capable of hardware handshake you can use 9600baud transfer rate. Use the following settings then:

On the PC-1600 type:

```
SETCOM"COM1 : ", 9600, 8, N, 1, N, N
INIT"COM1 : ", 512
SNDSTAT"COM1 : ", 59
RCVSTAT"COM1 : ", 61
OUTSTAT"COM1 :
```

Within the terminal program:

- Set binary (Bin) mode
- Activate CTS flow control (= hardware handshake)
- Deactivate DTR flow control (if that option exists)
- Deactivate XON/XOFF
- Baud rate: 9600, Data bits: 8, Parity: none, Stop bits: 1

If your connection is not capable of RS232 hardware handshake you should use the XON/XOFF protocol instead and a lower baud rate then (see PC-1600 manual).

The provided file SUPERRAM.BIN is a PC-1600 file, so you can directly upload it using this standard method. The load address `#0,&C0C5` is contained in the header, but no autostart address.

6.2 Upload via CE-158

Alternatively the upload also works with a CE-158 serial interface and the PC-1600 running in PC-1500 compatibility mode. You have to use the SUPERRAM.RAW (this file does not contain any header) if you use a PC software like lhcom resp. COM158 which builds the header, or SUPERRAM.158 which already contains a proper header. In PC-1500 mode, the SUPERRAM code will be installed at &40C5. In MODE0, the address is &C0C5.

- Plug the CE-158 serial interface to the PC-1600 and connect the CE-158 serial input and the host PC or MAC with a serial cable (it also works with a SERIAL/USB adapter, like FTDI).
- Switch the interface ON (Button on the top) and then the PC-1600. If you get a CHECK 8 message, you have to charge the batteries of the CE-158.
- Now you need to setup your PC-1600 to PC-1500 compatibility mode.
 - If you have not configured a main memory extension larger than 16KB you can just type:
MODE1
 - If you have configured a larger main memory extension (like with a CE-1600M in slot 1 INITIALIZED with "M" and/or the superRAM module in slot 2 INITIALIZED with option "M") the MODE1 command gives you an ERROR110. But there is a simple and important workaround. Just type:
POKE &F1BC,PEEK(&F1BC) OR 64
Now your PC-1600 is in PC-1500 compatibility mode regardless of your memory configuration!
- Type:
SETCOM 2400,8,N,1
SETDEV CI
CLOAD M
- On the host PC/MAC, use a transfer software, like COM158 or lhcom from the lhTools (<http://www.pc1500.com/lhTools>).
 - If your transfer tool knows how to build the serial header, use the SUPERRAM.RAW with the following options:
lhcom -s -Y =2400,8,N,1 -Zheader=CSAVEM -Zstart=40C5 SUPERRAM.RAW
This will build a HEADER of type BINARY, starting at address &40C5 and sends the binary image at 2400 bauds, 8-bits, no parity, 1 stop bits.
 - If your transfer tool is not able to build the serial header, use the SUPERRAM.158 instead.
lhcom -s -Y =2400,8,N,1 SUPERRAM.158
- After 18s the program is loaded
- Type:
MODE0

Now the SUPERRAM code has been loaded and you can continue with step 4 from the init procedure (see above).

6.3 Upload via CE-150

However if you do not own a CE-133T (or equivalent) nor a CE-158 but a CE-150, it is also possible to upload the program. You need the SUPERRAM.WAV file then. Here's how it works:

- Download, install and start Audacity
- Connect your CE-150 / PC-1600 to your host computer with the EAR/MIC cables of the CE-150
- Set the operating system mixer output volume for Audacity to about 80%
- Within Audacity
 - Open SUPERRAM.WAV
 - Check/set the correct audio device setting
 - Check/set output volume slider to MAX
 - Check/set the Project Frequency to 5000 Hz
- Now you need to setup your PC-1600 to PC-1500 compatibility mode (see above)
- Type:
CLOADM
- Start the audio playback within Audacity
- After 5-10 sec "SUPERRAM.BIN" must appear at the display, otherwise there is something wrong (cable connection error, wrong output device, volume too low, not in PC-1500 compatibility mode ...)
- Wait until the playback has finished and the cursor of the PC-1600 has reappeared.
- Type:
MODE0

Now the SUPERRAM code has been loaded and you can continue with step 4 from the Init Procedure (see above).

7 Using the RAM disk

Independently of the operation mode and the concrete size, the initialized RAM disk can be accessed using the following commands exactly as with the SHARP standard modules (for more information see the PC-1600 manual):

```
INIT, TITLE, DSKF, FILES, LFILES,  
LOAD, LOAD#, BLOAD, SAVE, SAVE#, BSAVE,  
COPY, KILL, NAME, SET,  
MAXFILES, OPEN, CLOSE, INPUT#, PRINT#, LOF, LOC, EOF
```

8 Memory Map

This section is intended for experienced developers and is only relevant for programming with ML / assembler.

The address range of slot 2 of the PC-1600 is &8000- &BFFF in bank 2 and 3 of the Z80 (2 banks 16KB each result in 32KB). In addition the *superRAM* module uses the so called vertical bank switch of slot 2 like the CE 1601M and the CE-1650M, in order to address more than 32KB in this area. The vertical bank switch is executed by write access on port 28H.

In Z80 assembler syntax:

```
LD  A,  #<vertical-bank>  
OUT (#28),A
```

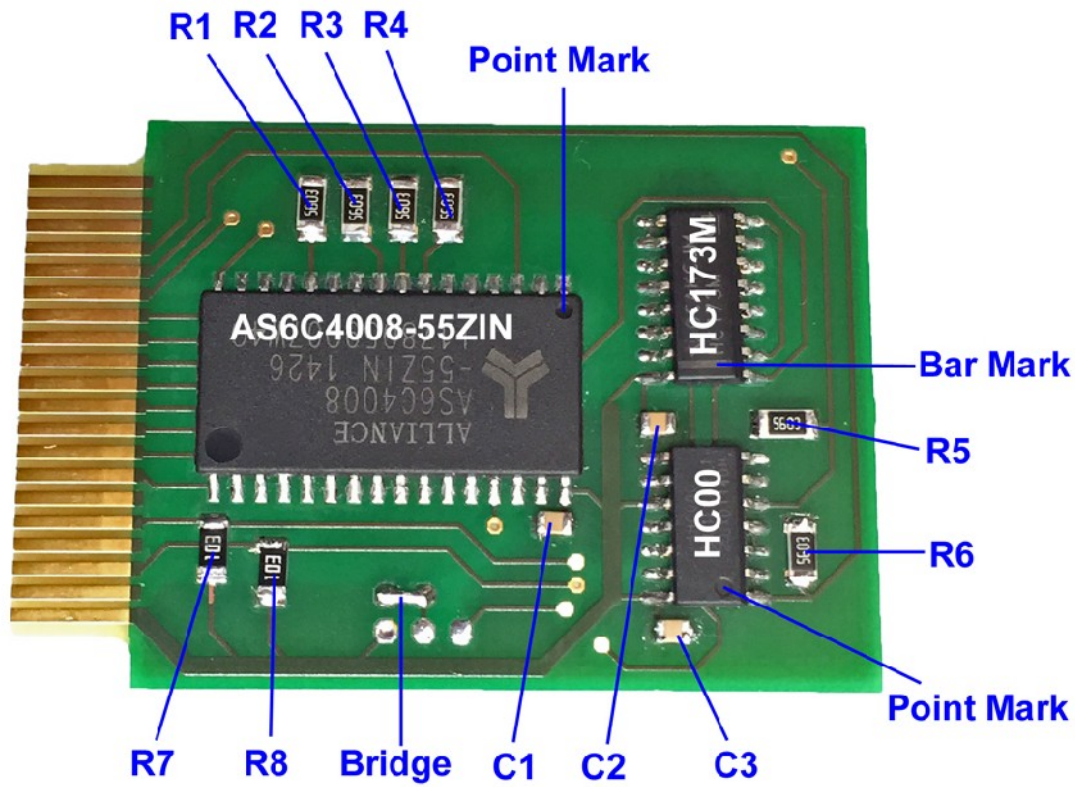
With the CE-1601M <vertical-bank> may take values 0 or 1, whereas in 256KB Mode it takes values between 0 and 7 (8 vertical banks 32KB each result in 256KB). So in 512KB Mode it takes values between 0 and 15.

The vertical bank switch is supported by the PC-1600 by slot 2 only, and only vertical bank 0 can be configured as main memory expansion (S0). The vertical bank switch is used transparently by the operating system of the PC-1600 to realize larger RAM/ROM disks.

For more information, see. 'SHARP PC-1600 Technical Reference Manual' (download e.g. at <http://www.sharp-pc-1600.de>)

9 Kit Assembly

The picture below shows the assembled board and the three ICs in correct orientation.



Besides the ICs the board layout provides some passive components included with the kit:
6 x **560KOhm** pull-down resistors: **R1 .. R6**
2 x **10KOhm** pull-up resistors: **R7, R8**
3 x **100nF** capacitors (one for each IC): **C1,C2,C3**

The board might/will run without these passive components - they are kind of convention/protection so you may leave them out, but if you like to be on the safe side, I recommend to put all components in place. If you plan to provide an additional buffer battery (see section 10) for data persistence when unmounted, at least the resistors are needed.

Details:

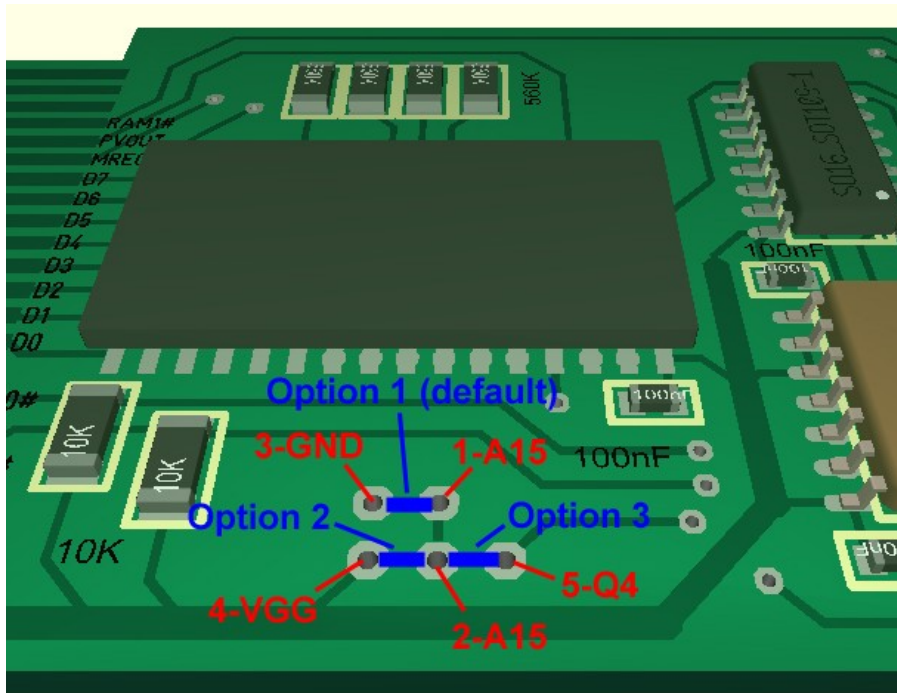
I adopted the pull-down/pull-up resistors from the CE-1601M which is a battery buffered module. That one has to take care of floating input signals when not mounted. The *superRAM* module does not (as long as you do not pimp it with a CR-20xx or something) and as far as I figured out the SHARP PC-1600 already takes care of defined input signals, even when switched off. Of most importance might be the 4 pull-down resistors connected to the address bus (R1, R2, R3, R4), because their signals are input to a 4-bit register IC.

NOTE:

The wire bridge shown in the picture fixes the 256KB Mode. If you want a different operation mode, you must provide pins for a jumper or an additional hardware toggle switch:

9.1 Pin/Pad Field

The heart of this module is a 4096Kbit SRAM IC with a 512Kx8 layout. Since the SHARP PC-1600 natively supports “only” 256KB RAM in slot 2, one address signal (A15) remains unused (i.e. fixed to LOW) by default. But the *superRAM* module supports additional configurations (see section 3) that utilize the A15 signal through the pad field shown in the picture.



You can either use a wire bridge for static configuration of Option 1 (256KB Mode) or micro jumpers with 2mm pitch - or you connect an external switch to the pads, especially if you want to use the 2x256KB Mode.

CAUTION:

- If you run the module with no bridge at all, you will have a floating input signal A15 on the address bus of the RAM-IC, which will result in unstable behavior of the module and increased power consumption.
- If you run the module with other or more bridges at a time than specified, you will have outputs wired to VGG or GND or even worse a short circuit between VGG (+4.7V) and GND which may destroy your PC-1600 and/or the module.

9.2 Pre-Mounting Tests

Before you test the assembled board within the PC-1600, you should at least test with a multimeter or similar equipment that there is

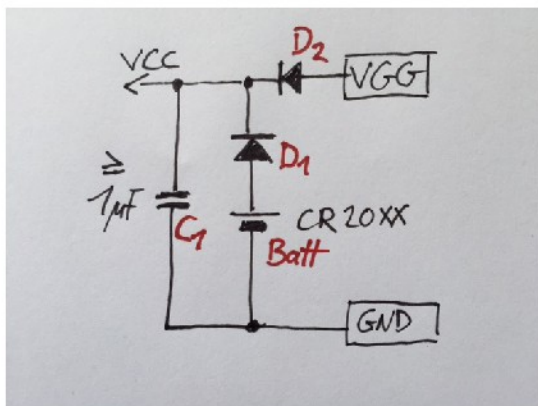
- no short circuit between VGG (+4,7V) and GND
- no bridge between adjacent IC-pins (except those bridged by PCB-design)
- electrical contact between each resistor- and capacitor-contact and the corresponding signal (IC-pin, VGG or GND)

10 Battery Buffer Modification

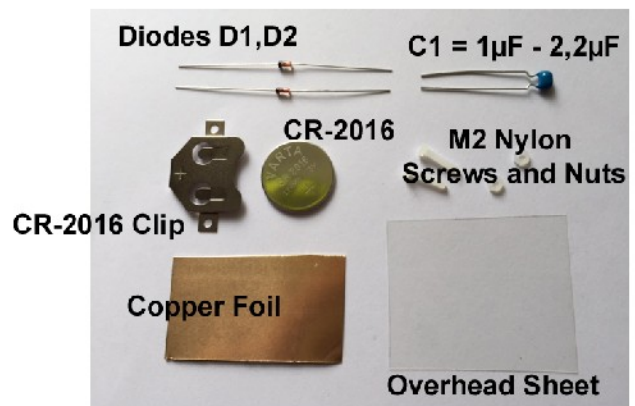
Despite the fact that the module design does not incorporate a battery buffer for permanent storage when unmounted, it is not very hard to make a suitable modification. I recommend using a CR-2016 which is very flat, so a standard clip holder for that battery type can be used with no rework (a CR-2016 clip holder is about 2.2mm high, that of a CR-2035 is about 4mm high, and the slot 1/2 compartment of the PC-1600 gives room for up to 3.3mm).

Have a look at the circuit diagram for the mod below: The capacity of C1 should be at least $1\mu\text{F}$. That gives you about 5sec for a battery change without losing data, which might be enough. A $2.2\mu\text{F}$ would give you about 10sec.

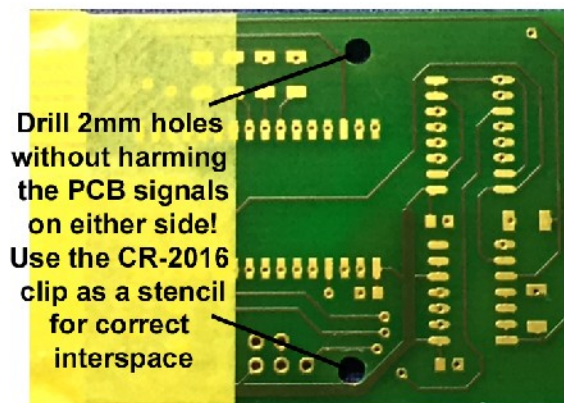
Circuit Diagram



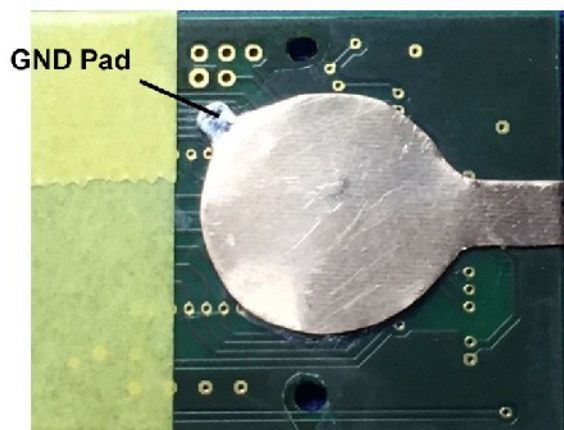
Parts Needed



Through Holes



Negative Terminal



Cut out from the copper foil a suitable (-)terminal for the CR-2016.

Prepare the GND pad with solder.

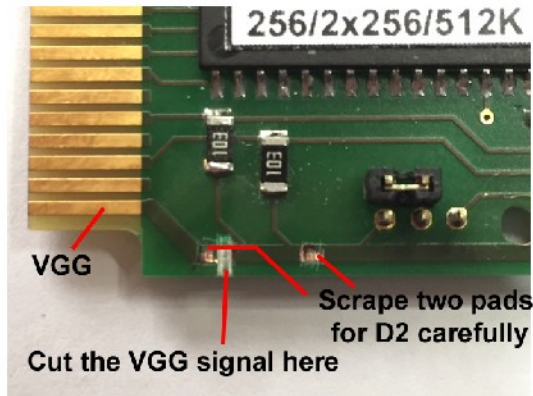
Cut out a suitable shape from the overhead sheet that leaves out the PCB jumper field.

Glue the terminal to the overhead sheet (e.g. epoxy).

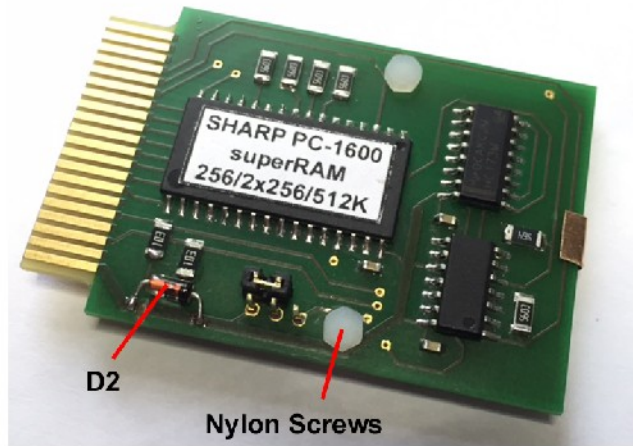
The terminal **MUST NOT** have any electrical contact with the PCB!

Now for the main assembly:

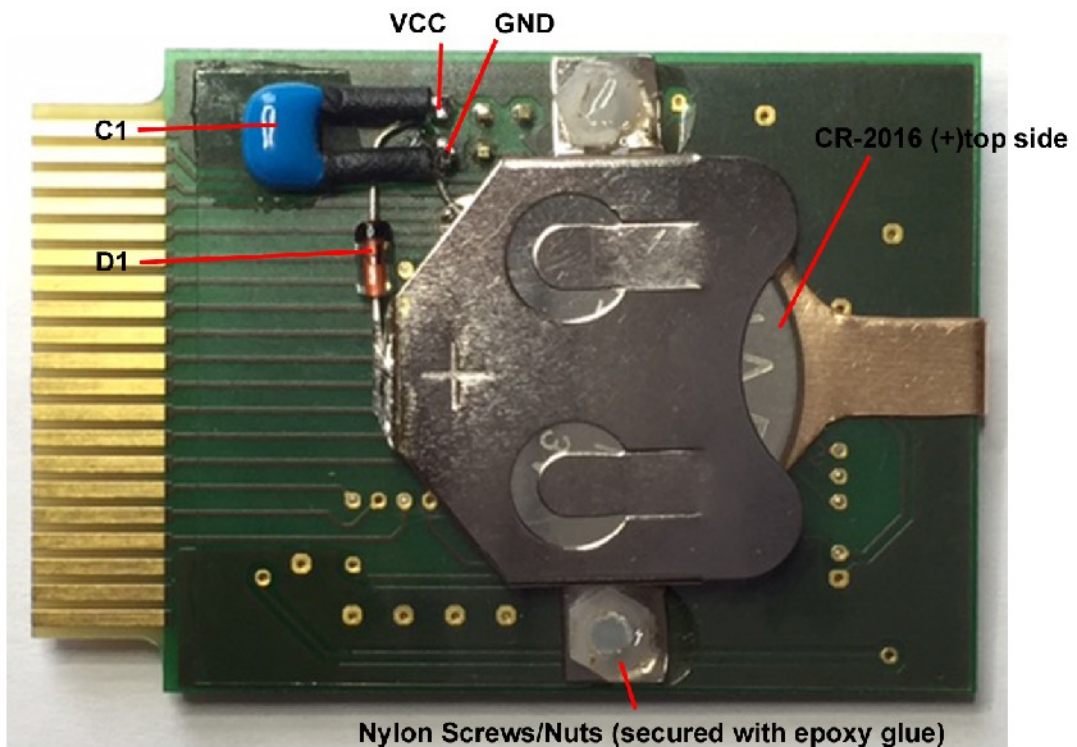
Prepare D2



Front Side Assembly



Backside Assembly



You may want to use an SMD capacitor instead, if you don't find a standard one that is flat enough.

Now this is a battery buffered 256/2x256/512KB module for your SHARP PC-1600 :-)