Updating the MB2K2 V1A PCB to V1B

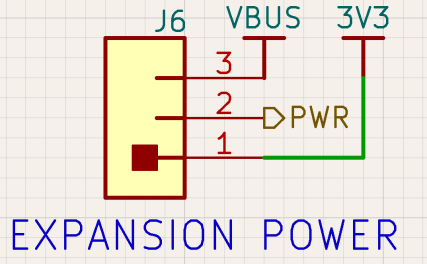
Introduction

The MB2K2 PCB has recently been updated from version 1A to version 1B. The new version has some minor changes from the old adding an I2C based non volatile F-RAMdisk and including the ability to route power to a spare pin on the FDC connector allowing it to be used as the basis of expansion boards.

This document covers the details of the upgrade process for the older version of the PCB.

FDC connector Power

The new PCB has a three pin header to allow either 3V3 or 5V to be routed to pin 4 of the FDC connector. This pin was chosen as it is unused in most implementations.

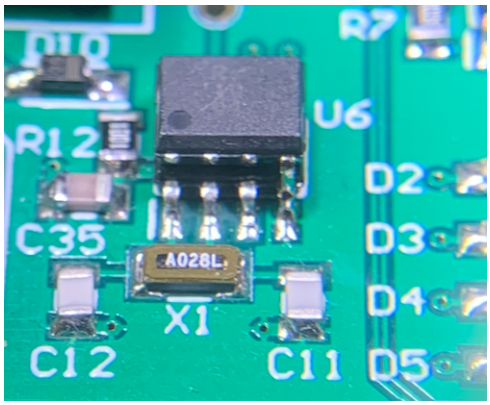


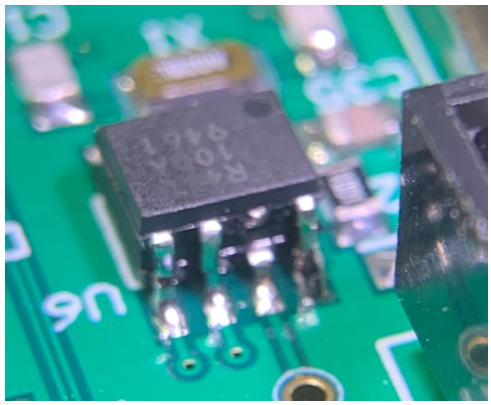
In practice it is simple to wire this pin to one of the power test points on the PCB to achieve the same result on the older PCB.

F-RAMdisk

The new F-RAMdisk part has been chosen with a view to simple upgrading of the older boards. It is an 8-pin SOIC part, the Fujitsu MB85RC1MTPNF-G-JNERE1 1Mb I2C based serial F-RAM and has been chosen for it's low price and availability (at the time of writing). In theory other I2C based parts may be used but be aware this may mean modifications to MON09 and the OS-9 ramdisk driver.

The simplest way to mount this part is by using the 'piggyback' technique where the new part is soldered on top of the existing RTC chip and although the result is not pretty, it works well in practice.





After removing the RTC battery, the F-RAM is prepared by carefully flattening the Gull Wing leads and bending them down so they are 90 degrees to the body. Pins 1, 2, 3 and 7 are then removed flush to the body leaving just pins 4 (GND), 5 (SDA), 6(SCL) and 8 (+V).

With this modification, the F-RAM power and signal pins are in the same places as the RTC and any potential clashes have been removed. The F-RAM has internal pull ups and pull downs that ensure the removed pins stay at the default signal levels.

Now position the F-RAM over the RTC pin 1 to pin1 such that the two device's pins touch, temporary secure with Kapton tape and after applying a touch of gel flux to each of the joints, solder the four sets of pins together.

Finally, check for opens and shorts.

Flashing the new firmware

Flashing the MB2K2's firmware requires the installation of the free Xmos xTIMEcomposer toolchain (version 14.4.1) and obtaining an 'XTAG' USB debug interface. The toolchain may be downloaded directly from Xmos at the tools download page and the XTAG interface is readily available from several suppliers world wide for about $20, see Octopart for details.

Documentation for the xTIMEcomposer tools is in the 'data sheets & app notes etc' folder in the firmware folder and covers importing project, the IDE, building the project and flashing. It may be also necessary to install the Java runtime

Download the latest MB2K2 release package (v0.95) from GitHub, unzip the project folder located in MB2K2_V0.95-release-master\firmware and locate the binary images in MB2K2\bin within the project.

Power on the MB2K2 and connect the XTAG to the board. (the XTAG may be safely connected or disconnected to a powered MB2K2) then open up the Xmos command line interface installed as part of xTIMEcomposer in the earlier step and use the following command:-

```
xrun -l
```

This command looks for the XTAG connector then reports on the connected device. The command should report something like :-

```
Available XMOS Devices
----------------------

  ID    Name                  Adapter ID      Devices
  --    ----                  ----------      -------
  0     XMOS XTAG-3           ZByhc2hJ        0[0]
```

if the command reports 'none' for the devices then there is a hardware fault with the board. If the power supplies and 24MHz clock are OK then this will most likely be an issue with a bad solder joint on the XU216 so check the pin connections and the ground connection under the PCB.

If all is well then the firmware may be flashed with the ‘xflash’ command thus:-

```
xflash --verbose --no-compression --factory <image.xe>
                    --boot-partition-size 0x100000 --data <promdisk.dsk>
where
<image.xe> is the path to the image file
           use MB2K2\bin\mb2k2_release_usb.xe as the default

<promdisk.dsk> is the path to the promdisk image
               use MB2K2/promdisk/MB2K2PD.dsk as the default
```

After about 30 seconds or so the flashing will begin with a series of messages of the form ‘site 0 write’ followed by an incrementing address as each flash block is written. With the current build and promdisk the block address will run up to about 0x003CFF00, (the actual number may be different to this as the build is updated).

After the flashing process is complete the MB2K2 will restart and you should see the debug LED (to the right hand side of the XU216) flashing on and off in a ‘breathing’ pattern. This is the sign that the firmware is working correctly. Power down and insert the RTC battery (noting polarity).

Using the F-RAMdisk

The F-RAMdisk is a new addition for the V1B PCBs and has no analogue to the original MB2 design. It is intended to free up internal Xmos RAM used by the existing RAMdisk thus creating space for larger processor emulation such as the 68K in future releases. The F-RAMdisk is non volatile and has no limitations on the number of writes (unlike flash) and needs no backup power so is ideal for a local scratch disk in cases where there is no remote storage connected via FlexNet.

Unlike the RAMdisk the F-RAMdisk isn’t formatted after power up and so must be formatted before first use. For Flex, this can be achieved either by the MON09 ‘DF’ command (40tr SS/SD only) or the ‘fmtfd' utility which can format the F-RAMdisk to either the standard Flex format of 40 track SS/SD (100KB) allowing the use of the standard Flex diagnostic utilities, or to a non standard format of 51 tracks SS/SD that maximises the available storage.

For OS-9 the usual ‘format’ command can be used.

For Flex the F-RAMdisk has a drive type of ‘2’ and the system utilities such as ‘allocate’ work in the same way as the other drive types. For OS-9, the F-RAMdisk appears as volume ‘f0’ and again works as the other mass storage types