

Compact Flash and SD Cards With Qubide Dilwyn Jones

Back in November, I spent some time tinkering with my Aurora-based Minis-QL computer, which is basically an Aurora motherboard mounted into a small Epson case with an Mplane backplane plus a Super Gold Card, ROMdisq and a Qubide IDE hard disk interface.



Minis-QL computer

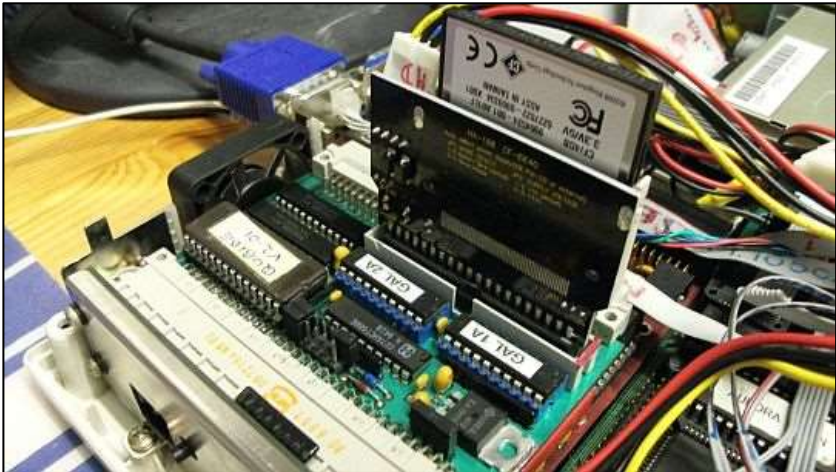
At the time, the old hard disk had failed and I was looking to fit a replacement. The machine was still usable via its 8MB ROMdisq flash memory card, but 8MB was a bit restrictive, even for a QL system!

I remembered reading somewhere that some Compact Flash (CF) memory card adaptors could be used via a Qubide, so I thought I'd try it out with a couple of compact flash adaptors I got from eBay and Amazon and much to my surprise it worked pretty much first time apart from an upside-down cable which luckily did no damage.

The picture below shows one of these Compact Flash adaptors plugged directly into a Qubide – although it worked first time, this made

the circuitry too large to fit in the Minis-QL case so an extension cable would have been needed to allow the adaptor to sit flat in another part of the case. On a Qubide, the connectors at either end of the qubide board are for the QL edge connector one end and a through connector for other cards at the other end. The drive cable connects to the male 40 pin IDE connector mounted on the circuit board, so these CF adaptor devices end up pointing vertically away from the Qubide circuit board.

The unit shown in the first picture is a single CF adaptor, the type you can get for a few Pounds on sites like Amazon. The second picture shows a dual Compact Flash adaptor. Some of these give you no choice which is the Master and which the Slave, while some have Master/Slave jumpers. I came across one which said it has a Master/Slave jumper, which on closer examination turned out to be only two solder pads you could bridge with solder!



A single slot Compact Flash adaptor plugged into Qubide



A dual-slot adaptor

Up to two of these adaptors could be added to a Qubide interface at a time, as a master-slave pair. The ones I used have a small pair of jumper pins on board which allowed the interface to work as a master unit, or as a slave unit, via a standard 40 pin IDE cable with two connectors on the end. 44 pin IDE connections can be used with Qubide, although the extra 4 pins carry power to 2.5 inch hard disk drives and flash memory adaptor boards like this, so since the Qubide uses 40 pin connectors, what you need is a small 40 to 44 pin adaptor which brings out the power pins to a separate floppy-disk style power connector. These usually come as either a small PCB with the relevant connectors attached, or as back-to-back 40 to 44 pin connectors with a flying lead to a floppy-disk style power connector, which you can just see to the left of the Compact Flash cartridge at the top of the first picture.



Example of one of the CF adaptors connected via cable rather than direct to the Qubide. Up to about a third of a metre of cable seemed to work OK.

I struck lucky and found a dual Syba CF adaptor which is cased to fit in a 2.5 inch hard disk drive and cost about £12.00. Where you need to mount the adaptor (as I did) in a space formally occupied by such a hard disk drive, this was extremely useful, although this device is of course a bit larger than the cards shown in the first two pictures. This device has a 44-pin IDE connection, so needs one of the power and 40-pin adaptors mentioned. One of the CF slots is on the top of the device, while the second slot is underneath. The device has screw slots to allow it to be mounted in a 2.5 inch drive bay.



The Syba 2.5 inch dual CF card adaptor, shown alongside a typical 2.5 inch hard drive. The Syba unit is shown with the small 40 to 44 pin connector adaptor PCB at the rear.



Syba Dual CF card adaptor in 2.5 inch hard disk form factor

My intention in using a dual slot adaptor is that when I have two CF cards connected, one can simply function as a backup or 'mirror' of the other. I have yet to write the background backup copier, which I hope will function in a similar manner to a screen saver – if the machine is inactive (no key presses or mouse activity for a while), files are copied if they have been modified since the last backup was made. This is necessary since the QL operating system does not really do background file copying – it takes over the machine and you can't do anything else until the copying has finished. More on my efforts in this direction in a future issue!

When it comes to formatting and using these flash memory cards via Qubide, it really is as simple as using a conventional hard disk. The main limitation is that of the FAT-based file system, that any partition can't be more than 2GB maximum, although multiple partitions are possible.

Version 2 of Qubide is really required to make this much use of these capacities, as it allows larger block sizes than the earlier versions. The FAT system allows for up to 65,536 blocks in a partition, so the smaller block size used, the smaller the maximum capacity. For example, if you use a 2K block size (block size '4' in Qubide-speak) , the maximum capacity of a partition becomes $(2K * 65536)/1024$ or 128 megabytes. If you use 32K clusters (block size 64 in Qubide) the maximum partition size becomes $(32K*65536)/1024$ or 2048MB or 2GB. The smallest possible file will always use one such block or cluster, so your file is only 1 byte in length, it may occupy the whole of a 2K, 4K, 8K, 16K or 32K block – obviously a trade-off between wasted space in each block and the maximum capacity of a partition.

You have to make the decision whether your system uses mainly smaller files, in which case you may be better off having more and smaller partitions on a memory card. If you regularly use larger files, consider fewer and larger partitions. Bear in mind also that the more partitions you have, the more memory in your computer will be used for the drive maps in memory, typically up to half a megabyte per partition.

To format the CF card, you can use the standard Qubide FORMAT command to create WIN1_. The command FORMAT WIN1_NAME (where NAME is a medium name of your choice) is used. When executed, it asks for the number of cylinders – enter a '0' to make it use the maximum possible. Then, enter the block size number from 4 to 64 as shown in the list below. Then sit back and wait. And wait...

As with hard disks, use the PARTITION_EXE program from the Qubide utilities disk to create further partitions. As I was using 4GB cards, I decided to create three 1GB partitions per card and the remainder of the 4GB (around 740MB) for the fourth partition, as in the table below. This should serve as a simple reference of how to calculate the various units and partition sizes. Cylinders are calculated a bit like you would the tracks on a hard disk. In this case I was using block sizes of 32.

Number	Name	Start	End	MB	Blocks
1	WIN1	0	2079	1023	32
2	WIN2	2080	4159	1023	32
3	WIN3	4160	6239	1023	32
4	WIN4	6240	7740	740	32

Qubide version 2 and later has a wider choice of block sizes to use than version 1, as shown here:

Block Sizes

4=2K clusters (max. 128MB partition)

8=4K clusters (max. 256MB partition)

16=8K clusters (max. 512MB partition)

32=16K clusters (max. 1024MB partition, or 1GB)

64=32K clusters (max. 2048MB partition, or 2GB)

When using a 4GB card, you may prefer to split it into two partitions of approximately 2GB each (actually, 2047 and 1764 MB respectively), using the settings shown below. This uses the maximum block size number 64, which builds the WIN drive in clusters of 32K.

Number	Name	Start	End	MB	Blocks
1	WIN1	0	4160	2047	64
2	WIN2	4161	7744	1764	64

Note that on my Minis-QL computer, it took up to an hour to format a 1GB partition. It varied a little depending on which adaptor and which CF card I used (some are presumably faster than others), but not very much variation. This became the single most trying part of the tinkering, I ended up letting it format overnight or while I was at work. Occasionally, it would have failed five minutes after I went to bed, for example.

You may have noticed in the example above that I used a 4GB card, with 3x1GB partitions and a 0.75GB one. These do not add up to 4GB! The reason being that the system uses some space which is prevented from being available to the user, and some hardware manufacturers tend to use 1GB=1000MB rather than 1GB=1024MB. What you should find happening is that the various partitions on a 4GB card will add up to about 3.7 or 3.8GB available to the user.

While assembling these and trying out these units, I bought several cheap samples of CF adaptors from various companies on the web. I found that not all would work together as master/slave pairs, although all but one (which eventually turned out to be faulty although I didn't know that at the time) worked as a single Master drive with no slave drive.

I learned the hard way that not all Compact Flash IDE adaptor boards are equal when it comes to Qubides: Some will not let you use more than 2GB in total and there was plenty of experience of the Partition_exe program locking up or just plain not working with some of the cheaper adaptors. I'm not sure if they skimp on the addressing or what else might be going on, but of the half dozen or so adaptors I have here, four worked perfectly, one was temperamental and intermittent (the faulty one mentioned above) and one would only allow me to create a single WIN drive and nothing else.

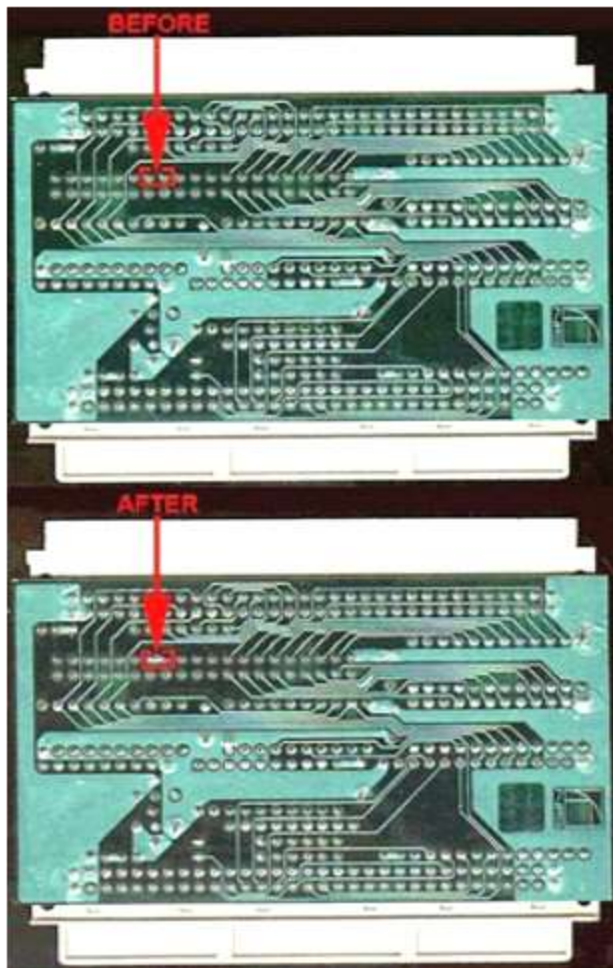
I'd happened to mention these issues on my blog and one or two people came up with various suggestions. Firstly, Bob Spelten mentioned a Qubide modification he had learned about from Peter Fox:

"In my previous comment on connecting the CF card I totally forgot to mention the Qubide modification I made. I wonder if you are aware of this tip that came from QLER Peter Fox:

– 'The modification affects only the ability to read CF cards and nothing else. I got this .jpg from Phoebus Dokos a very long time ago. I have got two IDE CF card readers attached to mine.' -

This may explain why some of your interfaces do work and others don't."

Bob later sent me pictures of the modifications to the tracks on a Qubide which help with some CF issues, although I don't know how they work, what they do, or what the risks to a Qubide are with these mods:



In case it's not too clear in print, the 7th and 8th tracks on the row of pins indicated are joined together.

Alain Haoui replied, describing how he has been using a hard disk, Zip 100MB drive, CF reader and IDE CD-ROM together on a Qubide. Yes, four units on an expander he built derived from an article by Nasta (Zeljko Nastasic, who designed the Qubide). Alain says that the hard part was that he had to adapt a major part of the Qubide driver (detection and timing issues, support of ATAPI devices like CDROM, ...) and in fact he rewrote the driver entirely to make it easier to

maintain and more compact to fit into the ROM. This work was in advanced step but not enough to make the new driver (called V3) available for the community. He says that as he has only one Qubide, he would welcome it if anyone has a spare Qubide they'd like to sell him.

Just when I thought I'd done just about everything I could with this, I stumbled on an extra item which may prove VERY useful.

While searching for a particular type of CF adaptor, I found that similar devices exist for SD cards. Several SD card interfaces have been devised for a QL, so I thought this could be very useful. I only found single SD card devices, and the prices were a bit higher than CF card units, and so far have been unable to get them to work as Master/Slave pairs, although they work fine as single units.

Like a CF card, the SD cards seem to format to Qubide standard. Now this cannot be used to directly transfer files between a QL and PC for example, since there is no suitable file transfer software I know of. But it does allow a Qubide user to have SD-card based mass storage quickly and easily.

The interesting part here is that the Qubide format is related to the Peter Graf/Adrian Ives format used for the QL-SD project, for example, although we do not yet know if compatible or how far the compatibility goes. Peter has kindly sent me files to test this when I get time and I'll report back on it in a future issue. There have also been some efforts to get code related to the format included in QemuLator and uQLx.



An SD card adaptor plugged into a Qubide – you can just see the 4GB SD card to the left of the silver casing on the black board. Note it has a hole in the PC for you to be able to get your fingers in to remove the SD card from its holder!

These CF and SD card adaptors are not the ultimate solution to the mass storage problem for a QL. They rely on having a Qubide unit, no longer in production and in short supply. They do offer a fairly quick and simple solution for the Qubide owner, though, until devices such as Peter Graf's QL-SD finally become available to buy.