The following problems are known to exist in the M212B version 0. Each is followed by a suggested means of getting round the problem. The version may be ascertained by issuing a **ReadParameter** of **M212Version (ErrorDrive)** immediately after reset in mode 1. This can be achieved by the following code sequence:

```
BYTE Version :
SEQ
  LinkToM212   ! #02 (BYTE); #23 (BYTE)
  LinkFromM212 ? Version
```

1.1 Hardware errors

1 The external memory interface signal **ALE** is normally high for one half of a processor cycle. However when an external cycle follows an internal cycle the **ALE** pin will go high one half cycle earlier making it asserted for a whole processor cycle. As the active edge of this signal is the falling edge the address setup and hold periods will be unaffected.
There is no real problem, but it may be necessary to be aware of it.

2 A glitch has been observed on the programmable ports. It occurs when the program does a read of the PIAPortData and is dependant on the last data byte written to the disk hardware. The glitch is a negative transition and the size of the glitch is dependant on the pin capacitance and the device temperature and supply. The glitch does not normally cross the TTL threshold and the effect can be minimised by increasing capacitative load or removed by modifying the program.

1.2 Software differences and errors

1 The default **Interleave** for winchesters is 8 instead of 6.
Interleave can be changed if required.

2 The floppy default values for **ECCPolynomial0/1/2/3** and **NumEccCorrectableBits** are all zero.
If using ECC for floppies then these parameters must be set up.

3 The minimum value for **NumBufferBytesBy256** is 5 instead of 1.
Make sure the value of **NumBufferBytesBy256** is greater than or equal to 5.

4 The maximum number of heads available for winchester drives is 8 instead of 16.
The **MotorOn** signal could be used as the extra head select bit but the head bits in the ID field will not be correct.

5 The amount of memory available for workspace when performing an auto-boot or a **Boot** command is 5 words less than specified.
Make sure that the program does not use the extra 5 words.

6 The **ErrorDrive** parameter is not implemented and will therefore remain set to the value of M212 version that is initialised at reset (i.e. 0).

7 A **BadPolyType** error gives a reason equal to the invalid mode bits as set in **Control** (i.e. #00 for chinese and #02 for disabled) instead of the mask of the mode field in **Control**.
If a **BadPolyType** error is encountered then **Control** will have to be examined to find out whether it is the ID or Data field mode that is in error.

8 The drive 0 hard parameters do not get initialised on reset.
Perform an explicit **Initialise** of drive 0.

9 A **DriveDoesNotExist** error may give a non-zero reason in the case where the **DriveExists** bit is not set.
No real problem, but it should not be confused with the **Initialise** error (see next problem).

10 If an **Initialise** command is given an invalid device type byte, then no **DriveDoesNotExist** error is flagged and **Reason** is not changed.
Make sure that only a #00 (floppy) or #01 (winnie) byte is given (see previous problem).

1.2 Software differences and errors

11 **FormatTrack** does not set a **FormatUnderrun** error if one occurs.
The status in the hard parameter **StatusRegister1** has to be examined after formatting each track.

12 The **BadDataCompareByte** bit in **Reason** is not set if a correction is attempted even if the byte was in error.
If it is necessary to find out if the data compare byte was incorrect after a correction has been attempted then the hard parameter **StatusRegister1** must be examined.

13 A class of uncorrectable errors are flagged as being correctable. These are where the error is indicated as being immediately before or after the actual sector. The chance of this type of error occurring is very small.

14 When in logical addressing mode with no auto-increment, **FormatTrack** still does an auto-increment.
The logical address should be reset after a **FormatTrack** command if required.

15 If doing a **ReadSector**, **WriteSector** or **FormatTrack** of a currently unselected drive, then the logical address calculation is performed using the selected drive's parameters.
Always issue an explicit **SelectDrive** command if any of the drives use logical addressing mode.

16 Performing an implicit drive select after having a drive selected which was not ready can cause a **DriveHasBecomeNotReady** error.
Always perform an explicit **SelectDrive** command.

17 During a multi-poll, if a floppy exists but is not to be polled then there will still be a motor start delay. With a drive without a Ready line this could take up to 1 second which is the poll timeout period.
Either reset the **DriveExists** bits of floppies before doing a multi-poll or poll each drive individually.