1.1 Audio Sculpture

More information about this program can be found in http://www.atari-forum.com/viewtopic.php?f=14&t=31939 . It is unclear if the diskettes for this program were duplicated on an Atari? However it seems that the protection used on track 3 required to manually modify some information on the diskettes after reading the duplicated diskette.

1.1.1 Track 0 + Tracks 4-58 Protections

All these tracks use very simple protections

1.1.1.1 Duplicate Sector Number (DSN)

All the tracks contains two ID field with a sector value of 66.

1.1.1.2 Sector with no Data (SND)

The two ID fields with value 66 do not have a corresponding data field.

1.1.2 Track 1 protections

1.1.2.1 Invalid Data in Gap (IDG)

The program is looking for the invalid character $F7 at the head of the track, that must be found before the first sync mark.

1.1.2.2 Hidden data into gap (HDG)

The program look for the specific sequence $92 $07 $91 $90 after the invalid character $F7

```
00016  1986 000541  E4 E4 E4 E4 E4 E4 E4 E4 E4 E2 82 14 7F 92 07 91   ääääääääää......
00032  1999 001032  90 90 FF FF FF FF FF FF FF FF C2 A1 A1 FE 01 00   ..ÿÿÿÿÿÿÿÿÂ¡¡þ..
```

Note the presence of the sync character before the $F7 that ensure that the sequence is always read correctly.

1.1.3 Track 2 Protections

This track has 5 x 1024 bytes sectors + 1 x 512 byte sector.

1.1.3.1 Hidden data into gap (HDG)

If we look at the beginning of the track we have:

```
00016  2000 000541  21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21   !!!!!!!!!!!!!!!!
00032  1984 001051  21 21 21 21 C2 0B CD B4 F7 00 DE AD FA CE 00 00   !!!!Â.Í´÷.Þ-úÎ..
00048  1984 001555  00 00 00 00 00 00 14 A1 A1 FE 02 00 01 03 37 26   .......¡¡þ....7&
```

We see that we have the following sequence: CD B4 F7 00 DE AD FA CE

And at the end of the track we have:

```
06176  1989 195483  08 08 08 08 08 08 08 08 08 08 00 31 00 48 61 0B   ...........1.Ha.
06192  1984 196002  CD B4 F7 DE AD C0 DE DE AF 4E 4E 4E 4E 4E 4E 4E   Í´÷Þ-ÀÞÞ¯NNNNNNN
06208  2000 196511  4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   NNNNNNNNNNNNNNNN
```

We see that we have the following sequence: CD B4 F7 00 DE AD C0 DE

1.1.3.2 Invalid Data in Gap (IDG)

In the hidden sequences described above, you can notice the presence of the invalid character $F7

1.1.3.3 Strange sector

This is not really a protection but the content of sector 6 is only “half used”?

```
05648  1986 179485  FB 4A D2 91 80 0E B1 FE 09 4A D6 80 7D 80 BC B0   ûJÒ...±þ.JÖ.}.¼°
05664  1984 179989  7B A1 88 E9 5A B1 79 27 34 81 40 ED 5A F5 7A D7   {¡.éZ±y'4.@íZõz×
…
05856  2000 186089  72 98 16 00 00 EB 4A 29 40 B5 0E 06 0E CC CE 61   r....ëJ)@µ...ÌÎa
05872  1984 186598  00 80 C2 29 40 CC CE 61 00 AB 9C 23 38 05 17 69   ..Â)@ÌÎa.«.#8..i
05888  1984 187107  B8 FF 0B FA CB 00 00 00 28 00 00 0B DE 2B 79 00   ¸ÿ.úË...(...Þ+y.
05904  1984 187614  CC 65 82 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   Ìe.ååååååååååååå
05920  2000 188125  E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   åååååååååååååååå
…
06128  1992 194754  E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   åååååååååååååååå
06144  2005 195264  E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   åååååååååååååååå
06160  1994 195778  E5 6F 8C                                          åo.
```

1.1.4 Track 3 protections

The track 3 uses potentially many protection mechanisms. But I am not sure which one are actually used / checked.

1.1.4.1 Write Splice Inside Sector (SIS)

The first and apparently impossible to reproduce protection (even on duplicator) is what I call: ***Write Splice Inside Sector (SIS)*** protection. The program definitively uses this protection.

Frédéric Batista (the author) describes the protection as follow:

The principle is very simple, it's based on the disk drives rotation speed variation on a whole round, it's in fact an extreme pain in the ass to replicate, maybe completely impossible.

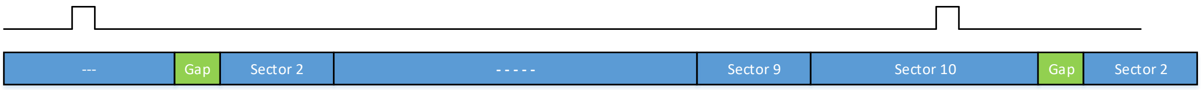
For that protection, you need to write a track that ends by the beginning of a long sector. Due to the speed variation, you never write twice the same exact track (i.e. the track write splice is never at the same location). When reading the last sector (which spread over the index pulse) you actually read data from the end of track plus data from the beginning of the track ***including*** *the write splice.*

By combining a "read track" command and a "read sector" command on the last sector, we can read the full circular content of the track. The content is always different for each diskette...

The only way to copy a diskette with this protection is to make a program that modify (on another track) an awaited value based on what is actually read on the newly copied track.

At a high level the track is written like this (note that the write splice happen in the Gap field):

But it is read like this:



Having the track write splice in the middle of sector 10 have potentially several effects when reading this sector:

- Sector 10 content will always be different each time the diskette is duplicated. However, on a specific copy the sector read the same (apart from exception described below).
- Having the write splice inside sector 10 can potentially create fuzzy bits. This behavior breaks the rule above, and therefore the protection need some tolerance on the expected content when reading sector 10.
- As the WD1772 sync mark detector is disabled when reading a sector, the content read at the beginning of the track is different from the one written during the write track.

Here is the beginning of the track as read by a read track command:

```
00000  1951 000030  67 E7 83 83 83 83 FF FF FF FB D5 25 41 32 A1 A1   gç....ÿÿÿûÕ%A2¡¡
00016  1984 000535  A1 49 6E 69 74 69 61 6C 20 50 72 6F 67 72 61 6D   ¡Initial Program
00032  1984 001038  20 6C 6F 61 64 20 5B 76 32 2E 30 5D 20 20 54 6D    load [v2.0]  Tm
00048  1984 001547  20 31 39 38 39 2C 31 39 39 30 20 54 68 75 6E 64    1989,1990 Thund
00064  1984 002062  65 72 53 6F 66 74 20 44 65 76 65 6C 6F 70 6D 65   erSoft Developme
00080  1984 002564  6E 74 C2 C2 C2 FE 50 00 01 02 63 6C F5 A1 A1 FE   ntÂÂÂþP...clõ¡¡þ
00096  1989 003072  50 00 0B 03 40 BE F7 A1 41 74 61 72 69 2D 53 54   P...@¾÷¡Atari-ST
00112  1984 003580  20 49 50 4C 20 54 72 61 63 6B 20 20 00 00 14 A1    IPL Track  ...¡
00128  2000 004090  A1 FB E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   ¡ûåååååååååååååå
00144  1979 004598  E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   åååååååååååååååå
```

Here the same region as read with the read sector command:

```
06284  1984 199727  20 20 20 20 20 20 20 20 20 20 20 F0 F0 F0 F0 00              ðððð.
06300  1994 200237  00 00 10 01 20 79 10 A0 A0 A1 20 08 20 18 20 E0   .... y.  ¡ . . à
06316  1979 200748  1C F8 78 48 00 80 48 E0 04 F8 18 00 E0 9C F8 00   .øxH..Hà.ø..à.ø.
06332  1984 201255  0C 4C 0C 78 04 FC F8 18 04 FC 64 24 3C 24 1C 64   .L.x.üø..üd$<$.d
06348  1984 201764  24 24 7C F8 18 38 00 08 98 80 48 40 00 88 1C F9   $$|ø.8....H@...ù
06364  1984 202275  98 80 08 80 18 00 78 00 80 08 11 41 41 40 08 7F   ......x....AA@..
06380  1972 202784  FF E7 C8 C0 10 00 A0 A0 08 7F FF 07 C1 F0 00 00   ÿçÈÀ..  ..ÿ.Áð..
06396  1984 203292  A1 E0 18 E0 48 24 00 40 1C F9 20 79 1C F8 18 48   ¡à.àH$.@.ù y.ø.H
06412  1984 203800  E0 C0 04 FC FF FF F0 A0 A0 A0 00 80 80 80 80 80   àÀ.üÿÿð   ......
06428  1992 204308  80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80   ................
06444  1984 204817  80 80 80 80 80 80 80 80 80 80 80 80 80 80 80 80   ................
```

As you can see some text (e.g. "Initial Program load") that are read correctly by the read track are not read correctly with the read sector command. This happen because the presence of sync marks in front of the text is used by the read-track command, while they are ignored by the read-sector command.

It is not clear how this protection is actually checked. Ijor believes that an algorithm like the one described below is used:

- Read sector 10 (that crosses the index and the write splice), and compute a checksum of the whole 1024 bytes of this sector.

- Read track, locate the first sync mark in the buffer (it is a few bytes after the index), and compute a checksum on all the bytes from that sync mark up to a few bytes before of the end of track.

Personally, based on what Frederic says, I would think that the checksum for the track is computed from location of sector 2 until sector 10 so that the full track is covered by combining data from the two read commands. But it really does not matter.

1.1.4.2 Hidden Data into Gap (HDG)

The beginning of the track contains three $A1 sync mark and therefore the bytes located after are always read correctly during the read track:

```
00000  1951 000030  67 E7 83 83 83 83 FF FF FF FB D5 25 41 32 A1 A1   gç....ÿÿÿûÕ%A2¡¡
00016  1984 000535  A1 49 6E 69 74 69 61 6C 20 50 72 6F 67 72 61 6D   ¡Initial Program
00032  1984 001038  20 6C 6F 61 64 20 5B 76 32 2E 30 5D 20 20 54 6D    load [v2.0]  Tm
00048  1984 001547  20 31 39 38 39 2C 31 39 39 30 20 54 68 75 6E 64    1989,1990 Thund
00064  1984 002062  65 72 53 6F 66 74 20 44 65 76 65 6C 6F 70 6D 65   erSoft Developme
00080  1984 002564  6E 74 C2 C2 C2 FE 50 00 01 02 63 6C F5 A1 A1 FE   ntÂÂÂþP...clõ¡¡þ
00096  1989 003072  50 00 0B 03 40 BE F7 A1 41 74 61 72 69 2D 53 54   P...@¾÷¡Atari-ST
00112  1984 003580  20 49 50 4C 20 54 72 61 63 6B 20 20 00 00 14 A1    IPL Track  ...¡
00128  2000 004090  A1 FB E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   ¡ûååååååååååååååå
00144  1979 004598  E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   åååååååååååååååå
```

1.1.4.3 Sector Data with no ID (SNI)

We can see that at location 129 we have the start of a data field ($FB preceded by three $A1 sync mark). However there no valid ID field in front of this data field. There is a pseudo ID field (sect=1) but it uses three $C2 sync marks (instead of normal $A1) and therefore it is not recognized as an ID by the WD1772. This pseudo-id field is followed by another strange pseudo-id field (sect=11) but again it does not contain a valid sync sequence ($F5 $A1 $A1) and therefore is also not recognized as a valid ID by the FDC.

1.1.4.4 Invalid data in Gap (IDG)

If you look again in the start of the track you will see that we have an $F5 at location 92 as well as a $F7 at location 102. These values cannot be written during a write track by the WD1772. Therefore, unless an unknown escape mechanism is used, the floppy could not be written on an Atari. Note that there is also a $F7 located at the end of the track (loc=6236).

1.1.4.5 Invalid track number (ITN)

All the ID fields in this track uses an invalid track number ($50). For example:

```
00688  1979 021930  20 20 20 20 00 00 00 00 00 00 14 A1 A1 FE 50 00       .......¡¡þP.
00704  1993 022441  02 02 EA 07 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E A1   ..ê.NNNNNNNNNNN¡
```

Here the track number is set to $50 (decimal 80). As the track number in the ID field is compared with the content of the track register, normally it is not possible to read this sector. However for reading this kind of sector is possible to play with the FDC track registers (i.e. load the track register with $50)).

1.1.4.6 ID within ID (IWI)

The ID fields for sector 2 through 9 are followed by a second ID field located (hidden) inside the gap of a normal ID field:

```
00688  1979 021930  20 20 20 20 00 00 00 00 00 00 14 A1 A1 FE 50 00       .......¡¡þP.
00704  1993 022441  02 02 EA 07 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E A1   ..ê.NNNNNNNNNNN¡
00720  1984 022952  A1 A1 FE 50 00 0C 03 D9 29 4E 4E 00 00 00 00 00   ¡¡þP...Ù)NN.....
00736  1978 023459  00 00 00 00 00 00 14 A1 A1 FB E5 E5 E5 E5 E5 E5   .......¡¡ûåååååå
00752  1986 023970  E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5 E5   åååååååååååååååå
```

If there was no second ID field inside the gap we would have:

```
00688  1979 021930  20 20 20 20 00 00 00 00 00 00 14 A1 A1 FE 50 00
00704  1993 022441  02 02 EA 07 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E
00720  1984 022952  4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 00 00 00 00 00
00736  1978 023459  00 00 00 00 00 00 14 A1 A1 FB E5 E5 E5 E5 E5
```

Here we can see an ID field with a sect=2 and length=2 (512 bytes) that is located 34 bytes (23 x $4E + 11 x $00) before the data field. As this ID field is positioned at an acceptable distance of the data field (between 32 and 43 bytes) it allows to read the sector normally.

But in reality, as we can see, this ID field is followed by a second embedded ID field with a sect=12 and a length=3 (1024 bytes). This ID field is located at only 13 bytes (2 x $4E + 11 x $00) before the data field. This gap is much too short for the WD1772 to be able to read the sector. However, it is possible to read this second ID field using a read-address command.

Note that for the last sector 10 we have:

```
05616   1984 178448   72 61 63 6B 69 6E 67 2E 2E 2E 2E 00 14 A1 A1 FE   racking......¡¡þ
05632   1984 178956   50 00 0A 03 73 8F 4E 4E 4E 4E A1 A1 A1 FE 00 14   P...s.NNNN¡¡¡þ..
05648   1976 179470   03 6A DE 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 00 00 00   .jÞNNNNNNNNNN...
05664   1984 179973   00 00 00 00 00 00 00 00 00 14 A1 A1 FB 53 4F 46   ..........¡¡ûSOF
```

The last sector is declared as sect=10 ($0A) and length=3 (1024 bytes). The second hidden id field has a sect=20 ($14), a length=3 but it has a bad CRC value.

To summarize we have 18 ID fields that can be read with read-address command. But only 9 sectors (8 x 512 bytes + 1 x 1024 bytes) that can be read with a read-sector command.

1.1.4.7 Strange sector

This is not really a protection but the content of the last sector before the index pulse contain somewhat unusual values:

```
05664   1984 179973   00 00 00 00 00 00 00 00 00 14 A1 A1 FB 53 4F 46   ..........¡¡ûSOF
05680   1993 180482   54 20 50 14 48 00 CC 0E 00 C9 08 08 67 CE 08 0C   T P.H.Ì..É..gÎ..
05696   1984 180990   CC 0C CE 41 20 67 C8 02 67 C0 C2 40 08 08 4C 27   Ì.ÎA gÈ.gÀÂ@..L'
05712   1992 181495   C8 4C 08 0E 27 C3 C7 02 06 07 00 E7 C0 CC 08 E0   ÈL..'ÃÇ....çÀÌ.à
05728   1984 182006   67 E0 E2 27 E2 20 67 E0 20 27 E1 23 27 E7 E7 E7   gàâ'â gà 'á#'ççç
05744   1984 182514   E7 E7 CC C9 02 08 27 C3 C2 48 00 CC 0E 00 C9 08   ççÌÉ..'ÃÂH.Ì..É.
05760   1984 183020   08 67 CE 08 0C CC 0C E7 CE 41 27 C8 02 67 CE 4F   .gÎ..Ì.çÎA'È.gÎO
05776   1996 183532   00 00 C9 02 00 CF 27 CC 42 44 04 C4 02 41 06 27   ..É..Ï'ÌBD.Ä.A.'
05792   1984 184039   C0 CC 08 E0 67 E1 22 67 E0 60 67 E0 62 27 E3 20   ÀÌ.àgá"gà`gàb'ã
05808   1981 184547   67 C1 04 67 C1 00 00 27 C7 02 44 27 C1 00 40 C4   gÁ.gÁ..'Ç.D'Á.@Ä
05824   1990 185053   02 44 02 00 E7 C6 41 27 C0 00 02 67 C3 C2 40 00   .D..çÆA'À..gÃÂ@.
05840   1984 185566   C4 06 00 E7 C2 00 20 60 60 7F E1 E1 14 28 14 14   Ä..çÂ. ``.áá.(..
05856   1989 186077   28 14 14 28 14 14 28 14 14 28 14 14 28 14 14 28   (..(..(..(..(..(
05872   1984 186584   14 14 28 14 14 28 14 14 28 14 14 28 14 14 28 14   ..(..(..(..(..(.
05888   1989 187095   14 28 14 14 28 14 14 28 14 14 28 14 14 28 14 14   .(..(..(..(..(..
05904   1999 187604   28 14 14 28 14 14 28 14 14 28 14 14 28 14 14 28   (..(..(..(..(..(
05920   1984 188112   14 14 28 14 14 28 14 14 28 14 14 28 14 14 28 14   ..(..(..(..(..(.
05936   1984 188624   14 28 14 14 28 14 14 28 14 14 28 14 14 28 14 14   .(..(..(..(..(..
05952   1984 189134   28 14 14 28 14 14 28 14 14 28 14 14 28 14 14 28   (..(..(..(..(..(
05968   1984 189641   14 14 28 14 14 28 14 14 28 14 14 28 14 14 28 14   ..(..(..(..(..(.
05984   2000 190152   14 28 14 14 28 14 14 28 14 14 28 14 14 28 14 14   .(..(..(..(..(..
06000   1984 190661   28 14 14 28 14 14 28 14 14 28 14 14 28 14 14 28   (..(..(..(..(..(
06016   1978 191168   14 14 28 14 14 28 14 14 28 14 14 28 14 14 28 14   ..(..(..(..(..(.
06032   2002 191679   14 28 14 14 28 14 14 28 14 14 28 14 14 28 14 14   .(..(..(..(..(..
06048   2000 192189   28 14 14 28 14 14 28 14 14 28 14 14 28 14 14 28   (..(..(..(..(..(
06064   1984 192698   14 14 28 14 14 28 14 14 28 14 14 28 14 14 28 14   ..(..(..(..(..(.
06080   2000 193210   14 28 14 14 28 14 14 28 14 14 28 14 14 28 14 14   .(..(..(..(..(..
06096   2000 193721   28 14 14 28 14 14 28 14 14 28 14 14 28 14 14 28   (..(..(..(..(..(
06112   1984 194230   14 14 28 14 14 28 14 14 28 14 14 28 14 14 28 14   ..(..(..(..(..(.
06128   2002 194742   14 28 14 14 28 14 14 28 14 14 28 14 14 28 14 14   .(..(..(..(..(..
06144   1992 195251   28 14 14 28 14 14 28 14 14 28 14 14 28 14 14 28   (..(..(..(..(..(
06160   1990 195759   14 14 28 14 14 28 14 14 28 14 14 28 14 14 28 14   ..(..(..(..(..(.
06176   2002 196272   14 28 14 84 50 14 61 C3 14 28 28 14 14 4C DA F7   .(..P.aÃ.((..LÚ÷
06192   2002 196783   4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   NNNNNNNNNNNNNNNN
```