AUFIT
User's manual

By Jean Louis-Guérin (DrCoolZic)
Revision 0.4 March 2014

Table of Content

Presentation

This document describes the usage of the **AUFIT** program. AUFIT is the acronym for ***Atari Universal Floppy-Disk Image Tool***. The program reads several Floppy Disk image's formats used with the Atari platform including images generated at the flux level by devices like SuperCard Pro, KryoFlux. The FD image is analyzed and its content is displayed using Graphical and Text output windows. It is also possible to write the Floppy Disk image using the Pasti .stx format used by many Atari emulators.

Usage has been designed to be as simple as possible for casual users. However the program also provides powerful features for advanced users.

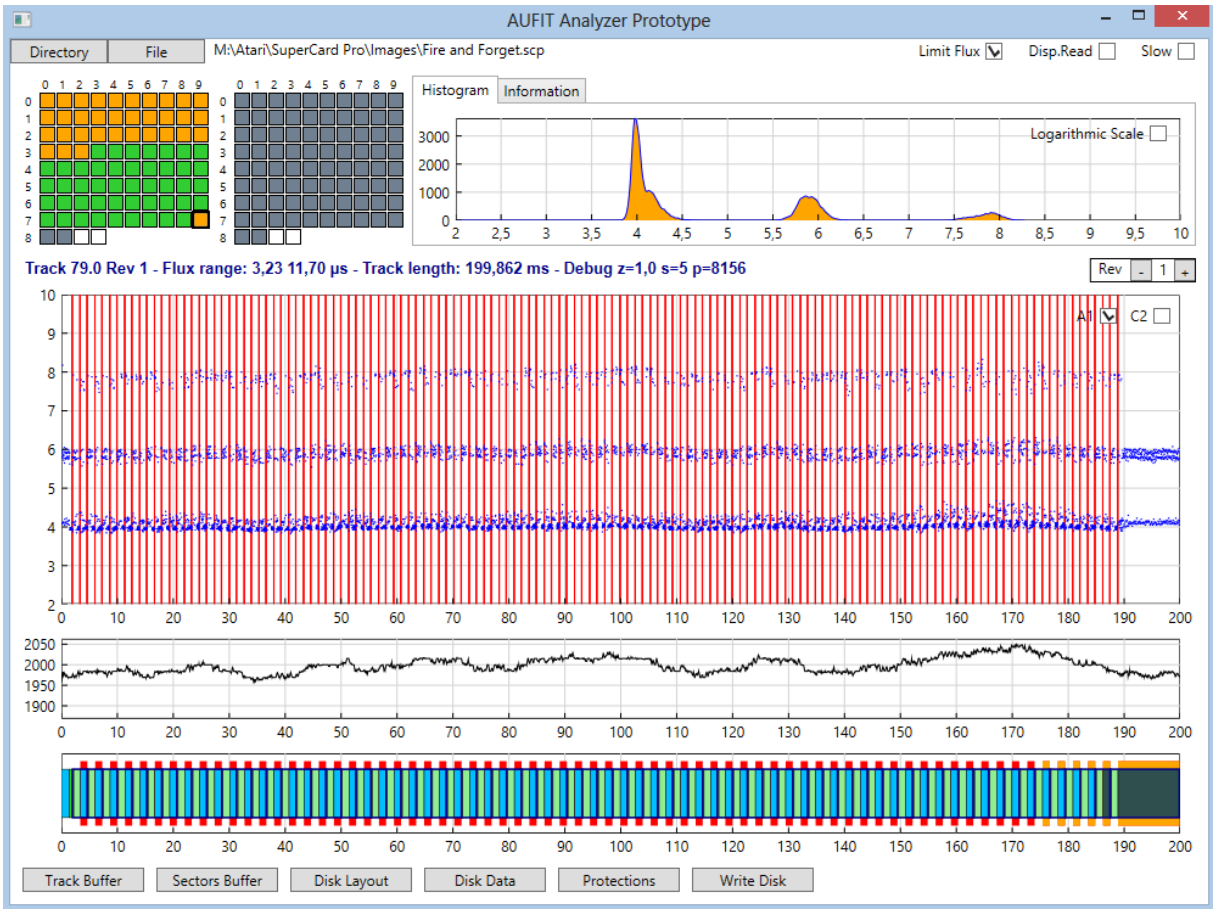
Beta Release Note

- The program is in beta test. Most basic features are now implemented, but the program may still terminate unexpectedly.
- If program terminate because running out of memory. Restart the program and try again.
- Currently most of the developments are focused on functionalities and therefore the GUI needs a lot of rework ☺

Running the AUFIT Program

Currently the program does not requires to be installed and is provided directly as an executable: **Aufit.exe**. To run the program just double click the executable.

The program run under Windows Vista SP2, Windows 7 SP1, and Windows 8 or above. The program requires that Microsoft.NET Framework 4.5 or above is installed. It also requires the Ookii Dialogs class library. The Ookii DLL is provided along with the executable and the documentation.



Reading Image Files

Currently the program can read a complete Floppy disk image composed of a set of **KryoFlux** stream files, or a **SuperCopy Pro** file.

To fully benefit from the ***Aufit*** analysis tools you need to sample a minimum of at least **three** revolutions for each track but **four** is the recommended value. Please refer to the tools documentation to create FD images with 3 or more revolutions.

Reading Kryoflux set of FD stream files

You first need to use DTC to save the content of a complete floppy disk as a set of raw stream files in a directory. To read all the stream files in ***Aufit*** you must click the **Directory** Button. A form open and allow you to specify the location of the directory that contains the raw stream files. The stream files inside the directory can have any name but all of them must have the same "**root**" name (for example awesome) followed by .Track_Num followed by .Side followed .raw.

For example *awesome.00.0.raw*, *awesome.00.1.raw*, *awesome.01.0* etc. Reading all the files (usually 170 files) is IO intensive and may take some time.

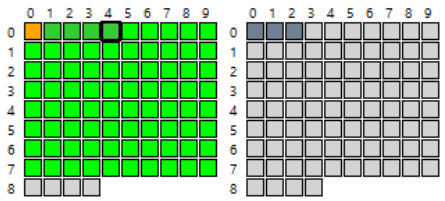
SuperCard Pro FD image

You first need to use SCP to save the content of a complete floppy disk as one big file with the extension .scp for example *awesome.scp*. To read the content of the SCP Floppy disk image you must click the **File** button and select the .scp file to open.

Kryoflux Single file

It is also possible to read a unique stream file using the **File** button. However beware that that several commands in ***Aufit*** only make sense on a complete disk image. For example displaying the Disk Layout.

Track selection using the track grid



Once the FD image is loaded the information for the first track of the disk is displayed in the different graphs. The program also displays two track grids composed of selection buttons: one grid for each side of the floppy. At this stage a quick analysis of the track content is done and is used to determine the background color of the buttons:

- *light green* for normal tracks,
- *light gray* for unformatted tracks, and

As soon as a track is fully analyzed the background color will turn to:

- *Dark green* for normal tracks,
- *Red* for protected tracks,
- *Blue* for track with no sector that has data that can be read (3+ sync),
- *Yellow* for track with possible data but hard to read,
- *Dark gray* for unformatted/empty tracks

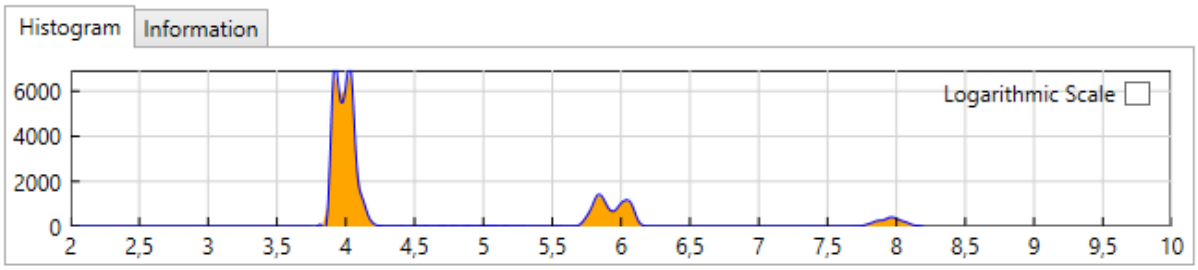
The currently selected track button can be identified by larger black border around the button of the grid. Leaving the mouse over a button display the track number.

If you click on a button of the track grid this will cause the track to be fully analyzed and all the graph to be updated for the selected track. The background color is therefore also updated as explained above. Once the track grids are "active" (for example by clicking one track selection button) you can move from one button to its neighbor by using the ***direction keys*** of your keyboard.

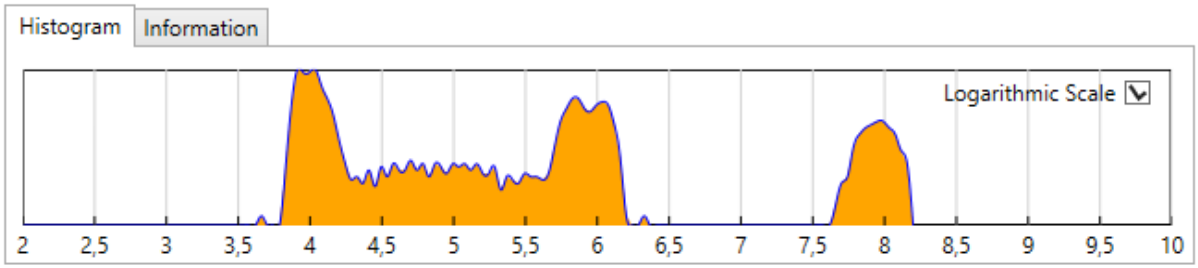
Note that full analysis of tracks is done whenever information about a track needs to be displayed. Therefore the **Disk layout**, **Disk Data**, **Protections**, and **Write Disk** buttons will force the analysis of all tracks. Once analyzed a track does not need to be reanalyzed (as the analysis information is saved) and therefore many operations execute faster once all the tracks have been analyzed (for example **Disk Layout**).

The Histogram Graph

This histogram representing the flux transition repartition and is displayed at the right of the button grid. Normally you should see three pics at 4, 6, and 8 microseconds.



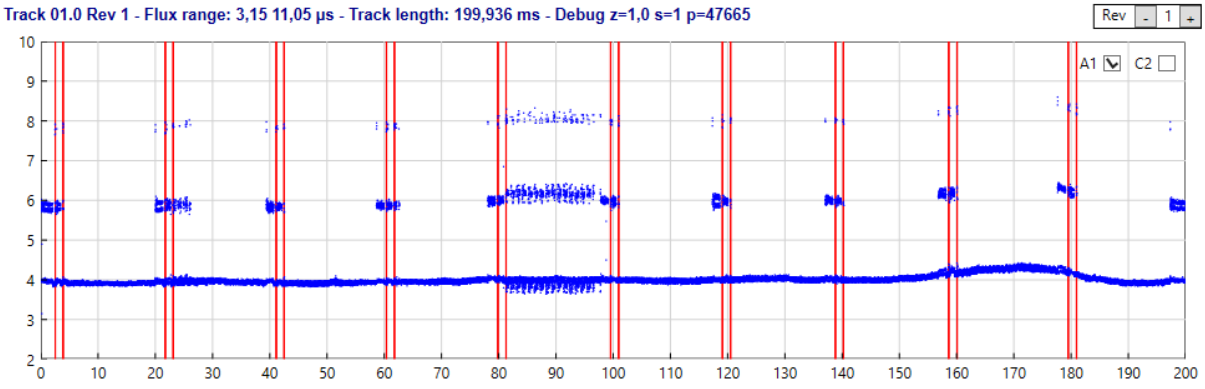
It is sometimes interesting to click the "**Logarithmic Scale**" check box to switch to a logarithmic scale on the Y axis. This allow to see information not visible on a linear scale. For example with the game Dungeon Master you will see that the histogram has a lot of entries between 4 and 6 microsecond due to the "sliding fluxes" pattern used as a protection for this game.



Flux Transitions Chart

The unprocessed flux transitions (i.e. flux directly read from the file) are displayed in a ***scattered chart***. A standard track is about 40000 flux transitions. To speed up the display time only 10000 flux transitions are displayed by default. If you want to display all the flux transitions you can click the **slow** check box in upper right corner of the window, this will force all the flux transitions to be displayed but the display will be slower. In the upper right of this chart you have two check boxes called ***A1*** & ***C2***. These check boxes are used to enable/disable the display of the **0xA1** and **0xC2** Sync Marks respectively as they would be detected by the WD1772 using a ***Read Track*** command (i.e. with the sync detector *activated at all time*). You need to refresh the display to see or hide the sync marks.

By default the **A1** sync marks are displayed and the **C2** sync marks are not displayed.



There is a check box button in the upper right corner of the program window called **Limit Flux**. By default this button is checked for unformatted track and is unchecked for formatted track but you can change its state at all time. Both the Histogram chart and the Flux Chart use this information and there content will be updated accordingly.

This is very useful when the track contains long flux transitions (like with NFA) to keep the graph readable.

Information about the currently selected track are displayed in a text box with the following information: track number, side, revolution, min and max value of the flux transition range, the track length, and some debug info.

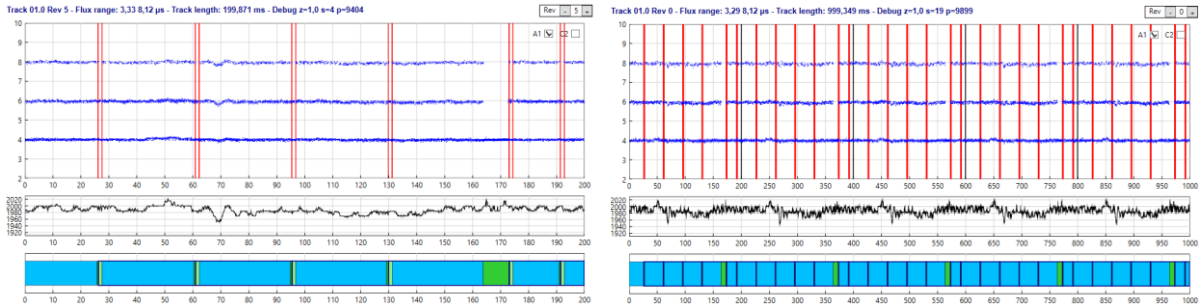
Selecting Revolution Data

The box on the right indicates the currently selected revolution. You can use the **+** and **-** button to select the data to display among the different revolutions sampled.

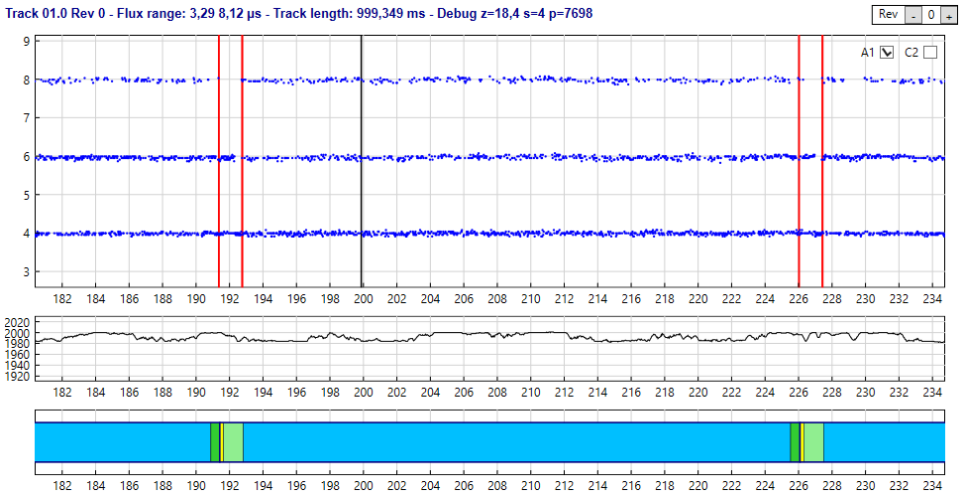
Selecting a specific revolution affect all displayed information including the ***Track Buffer*** and ***Sector Buffer*** display.

One very nice feature is that if you set the revolution number to 0 then the transitions for **all** sampled revolutions are displayed! The transitions from different revolutions are separated by black lines. Of course the x timing axis is extended to display all revolutions. Be aware that if you check the **slow** button while displaying the data for all revolutions the display will be very slow as about 200000 points needs to be displayed.

This feature is nice to see "shifted track" protections like data over index, data beyond index, ID over index (and even Sync over index!). For example in Turrican track 2 we have the last sector that starts close to the end of the track.

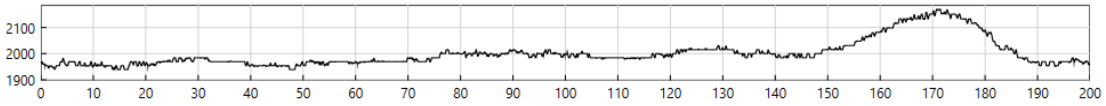


On the second chart we can see that the last sector starts close to the end and continue on the next revolution. If we now zoom we can see clearly what is happening (revolutions ends at 200 ms and is separated by a black line).



Clock Chart

The clock chart displays the period as computed by the internal DPLL clock. The nominal clock period is 2000ns but of course this value can vary a lot for over / under clocked sectors / tracks.



This clock is computed by the DPLL and represent a running average frequency of the flux transitions fed to the DPLL. The Y scale is automatically adjusted.

The clock should normally stay relatively stable around the 2000ns period but for protection like Copylock the period usually goes from 2000 to 2100 (5% increase) inside a sector. Some protections over clock or under clock the complete track.

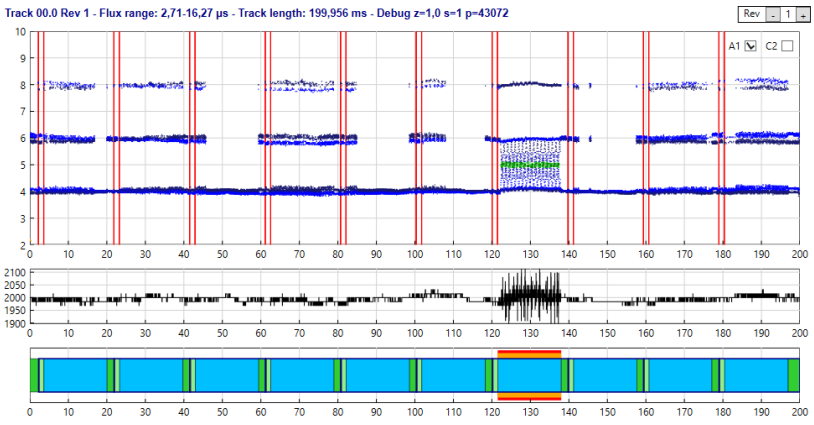
In some specific cases like unformatted track/sector the period can go up to 2200ns or down to 1800ns, reaching the 10% limits preset in the DPLL.

Data Graph

This graph display the data as they would be read by the ***Read Track*** command of the WD1772. The following color are used:

- The ID field is displayed in yellow (you probably need to zoom to see it),
- the Intra-gap field is displayed in light green,
- the Data field is displayed in sky blue, and
- The Inter-GAP field is displayed in lime green.

The data field color can change to aqua (lighter blue) for under clocked sector, to dark blue for overclocked sectors, and to blue violet for sector with intra byte bit width variation (e.g. speedlock).



Data sectors with CRC error have a red border, and data sectors with fuzzy bits have an orange border.

If you place the cursor over a sector in the data graph detailed information about the sector is displayed. If you double click a sector the content of this sector (as read by the FDC with a ***read sector*** command) is displayed in a separate window.

The "track data" information displayed is processed by a complex algorithm (i.e. it is not a simple dump of a ***read track*** command). This allow to display correctly "complex" tracks containing protection like: Sector within sector, tracks with large number of sectors, tracks with one sector, etc.

Zooming and Panning in the Charts

When the **flux chart** is not zoomed (default when first displayed) you can use the left mouse button to define a zoom area (zoom in both in x and y directions). Click on a first point to start the definition of the area, keep the mouse button down and move to a second point to define a zoom area. A rubber band box is drawn that delimitates the area and when the left button is released the graph is zoomed in the area defined. The scales displayed on both axis are automatically adjusted function of the zoom factor.

Once zoomed a click with the left button allow to pan the graph. Click any point in the flux graph with the left button and while keeping the button down move the cursor. The image of the graph dynamically moves with the cursor. When you release the mouse button, the graph and the axis position are updated.

You can further **zoom-in** or **zoom-out** by using the mouse wheel.

If you want to reset the graphs to un-zoomed state click anywhere in the Flux chart with the right mouse button (this is also the way to force a redisplay).

Note that the same zoom factor is used to update the **flux chart**, the **clock chart**, and the **data chart**. Therefore these three graphs always stay synchronized.

✎ Note that zooming in the Y direction only happen in the flux chart.

✎ You must place the mouse in the **flux chart** to pan and zoom any of the charts.

Track Buffer

If you click the **Track Buffer** button a new window will open that display several information about the currently selected track for the defined revolution.

Track info

Contains the track/side number, the number of bytes, the track length as sampled by the imaging tool, and the number of sectors.

```
Read Track for track 00.0 6284 bytes Length 199840 µs with 10 sectors
```

Track layout info

An array containing one line for each sector of a track, as well as a pre first sector that contains the GAP and eventually DATA segment information.

ID				INTER-GAP		DATA					INTRA-GAP	
SCT	POS	LEN	CRC	BYTE	LEN	BYTE	POS	LEN	CRC	FUZ	BYTE	LEN
											57	1887
1	1887	222	OK	34	1179	515	3289	16433	OK	No	25	1179
2	20582	222	OK	35	1196	515	21982	16397	OK	No	24	1196
3	39237	222	OK	34	1174	515	40635	16374	OK	No	25	1174
4	57869	223	OK	34	1178	515	59270	16386	OK	No	24	1178
5	76514	222	OK	34	1178	515	77915	16429	OK	No	24	1178
6	95203	222	OK	34	1178	515	96604	16448	OK	No	25	1178
7	113909	222	OK	34	1174	515	115306	16388	OK	No	24	1174
8	132557	223	OK	34	1178	515	133959	16404	OK	No	30	1178
9	151224	223	OK	34	1179	515	152627	16381	OK	No	24	1179
10	169869	222	OK	34	1180	515	171273	16382	OK	No	378	1180

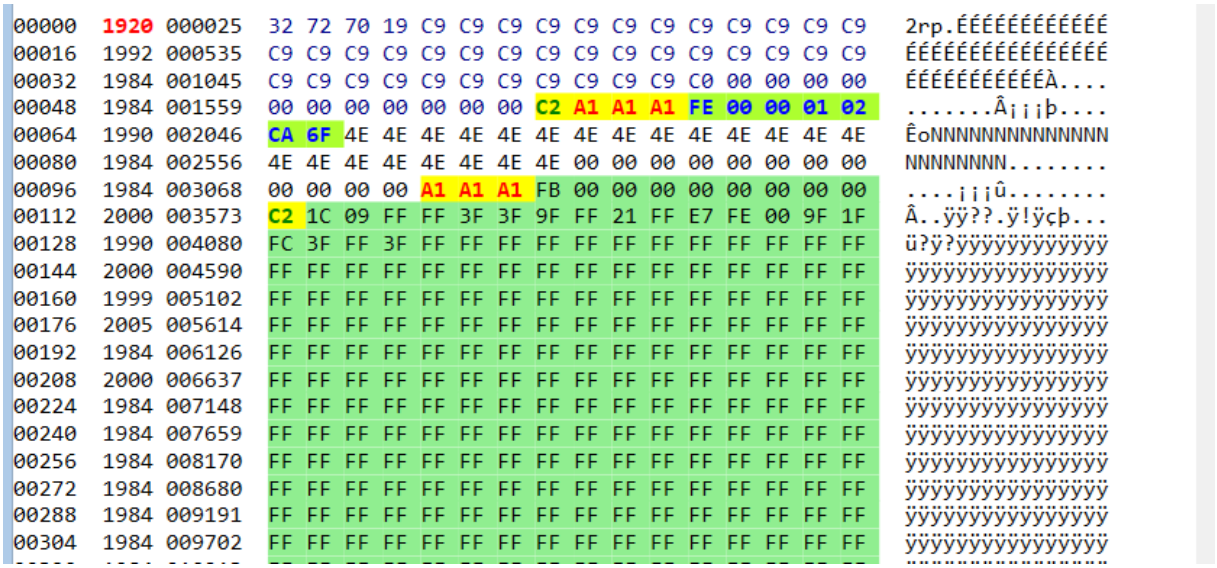
For each sector the following information is available:

- ID field: The sector ID, the position of the ID in microseconds from the index, the length of the ID field in microseconds and if the CRC of the ID is correct.
- Inter-Gap field: The number of bytes and the length of the gap in microseconds.
- Data Field: The number of bytes as decoded from the ID segment, the position in microseconds from the index, the length in microseconds, two indicators to display the status of the CRC and if the sector contains fuzzy bytes.
- Intra-Gap field: the number of bytes to reach the next sector and the length in microseconds.

Track content

A dump of all the data bytes exactly as they would have been read by the ***Read Track Command*** of the WD1772. The data bytes are displayed by group of 16 with the following information:

- The first number is the position of the byte in the buffer (decimal value)
- The second number is the period of the first byte of the line. If the period is 2.5% above normal the period is displayed in blue, if the period is 2.5% below normal it is displayed in red.
- The third number is the starting position of the first byte of the line in microsecond relative to the index of the track.
- 16 bytes are displayed in hexadecimal format
- The corresponding 16 ASCII characters for the 16 bytes.



```
00000  1920 000025  32 72 70 19 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9   2rp.ÉÉÉÉÉÉÉÉÉÉÉÉ
00016  1992 000535  C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9   ÉÉÉÉÉÉÉÉÉÉÉÉÉÉÉÉ
00032  1984 001045  C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C0 00 00 00 00   ÉÉÉÉÉÉÉÉÉÉÉÀ....
00048  1984 001559  00 00 00 00 00 00 00 C2 A1 A1 A1 FE 00 00 01 02   .......Â¡¡¡þ....
00064  1990 002046  CA 6F 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E 4E   ÊoNNNNNNNNNNNNNN
00080  1984 002556  4E 4E 4E 4E 4E 4E 4E 4E 00 00 00 00 00 00 00 00   NNNNNNNN........
00096  1984 003068  00 00 00 00 A1 A1 A1 FB 00 00 00 00 00 00 00 00   ....¡¡¡û........
00112  2000 003573  C2 1C 09 FF FF 3F 3F 9F FF 21 FF E7 FE 00 9F 1F   Â..ÿÿ??.ÿ!ÿçþ...
00128  1990 004080  FC 3F FF 3F FF FF FF FF FF FF FF FF FF FF FF FF   ü?ÿ?ÿÿÿÿÿÿÿÿÿÿÿÿ
00144  2000 004590  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00160  1999 005102  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00176  2005 005614  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00192  1984 006126  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00208  2000 006637  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00224  1984 007148  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00240  1984 007659  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00256  1984 008170  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00272  1984 008680  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00288  1984 009191  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
00304  1984 009702  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF   ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

Colors and font style are used to provide additional information.

Background color:

- Light blue for invalid characters (F5-F7) in gap
- Coral for no flux bytes (MFM 0x0000)
- Yellow for Sync characters (MFM 0x4489 and 0x5224)
- Light green for bytes located inside an ID field or a Data field.

Character color:

- Red for 0x4489 Sync characters (0xA1)
- Green for 0x5224Sync characters (0xC2)
- Blue for ID bytes
- Black for Data and intra-gap byte
- Black for inter-gap byte

Bold characters:

- For Sync characters (MFM 0x4489 and 0x5224)

Font Style

- Italic: When shifted track are used (data over index, data beyond index …) the read track command is extended to the beginning of the next revolution so the data are interpreted correctly. In that cases data beyond the index are displayed using italic style.

Sector Buffer

If you click the **Sector Buffer** button a new window will open that display the bytes exactly as it would have been read by combining the ***read address*** and ***read sector*** commands of the WD1772.

Sector info

Contains the track/side number, and the number of sectors.

```
Read Sectors for Track 00.0 has 10 sectors
```

Sector layout info

An array containing one line for each sector of a track.

For each sector the following information is available:

- ID field: The sector ID, the position of the ID in microseconds from the index, the length of the ID field in microseconds and if the CRC of the ID is correct.
- Data Field: The number of bytes as decoded from the ID segment, the position in microseconds for the index, the length in microseconds, two indicators to display the status of the CRC and if the sector contains fuzzy bytes.

```
--------------------+---------------------------
ID                  | DATA
SCT    POS LEN CRC  | BYTE    POS    LEN CRC FUZ
--------------------+---------------------------
  1   1887 222  OK  |  515   3289 16433  OK  No
  2  20582 222  OK  |  515  21982 16397  OK  No
  3  39237 222  OK  |  515  40635 16374  OK  No
  4  57869 223  OK  |  515  59270 16386  OK  No
  5  76514 222  OK  |  515  77915 16429  OK  No
  6  95203 222  OK  |  515  96604 16448  OK  No
  7 113909 222  OK  |  515 115306 16388  OK  No
  8 132557 223  OK  |  515 133959 16404  OK  No
  9 151224 223  OK  |  515 152627 16381  OK  No
 10 169869 222  OK  |  515 171273 16382  OK  No
```

Sectors content

For each sector a line that indicates the sector number, the number of bytes for this sector, the position in microseconds from the index, the length in microseconds, indications to say if the CRC is good or bad, and if the sector has fuzzy bytes.

This is followed by a dump of all the data bytes exactly as they would have been read by the ***read sector*** command of the WD1772. The data bytes are displayed by group of 16 with the following information:

- The first number is the position of the byte in the buffer (decimal value)
- The second number is the period of the first byte of the line. If the period is 2.5% above normal the period is displayed in blue, if the period is 2.5% below normal it is displayed in red.
- The third number is the position of the first byte of the line in microsecond relative to the index of the track.
- 16 bytes are displayed in hexadecimal
- The corresponding 16 ASCII characters for the 16 bytes.

```
Sector 1 515 bytes start at 3289 length 16433 Good CRC
00000  1984 003289  FB 00 00 00 00 00 00 00 00 A5 0D 90 00 02 02 01   û........¥......
00016  2000 003803  00 02 70 00 40 06 F9 03 00 0A 00 02 00 00 00 00   ..p.@.ù.........
00032  2000 004314  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00048  2000 004826  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00064  2000 005338  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00080  2000 005850  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00096  1984 006361  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00112  2000 006872  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00128  1984 007383  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00144  1984 007895  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00160  1984 008405  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00176  1984 008915  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00192  2000 009426  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
```

If fuzzy bytes have been found for one sector a "fuzzy mask" buffer is also displayed.

To find fuzzy bytes the Aufit program read the same sector multiple times from the different revolutions. If the content of a byte in the same position is different the mask value is the result of a **xor** operation.

```
Mask[offset] = (Sector_read_1[offset] ^ Sector_read_2[offset])
```

Thus same value (1&1 or 0&0) gives a result of zero and different value (0&1 or 1&0) gives a result of one. For example if the same byte is read one time as 0000 1111 and the next time as 0101 0101 the mask will be 0101 1010.

The mask can then be used to emulate fuzzy bits as follow:

```
FuzzyByte[offset] = (Sector[offset] & Mask[offset]) | (rand() & Mask[offset])
```

Where rand() is a function that randomly returns true or false.

The information for a sector presented above repeats for all sectors.

```
Fuzzy bytes for sector 7
00000  1977 121450  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00016  1984 121960  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00032  1994 122468  00 00 80 00 00 00 00 80 80 80 00 00 00 00 00 00   ................
00048  1984 122977  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80   ................
00064  1967 123486  80 00 00 00 00 00 80 40 00 00 00 00 00 00 00 00   .......@........
00080  1984 123996  00 00 00 00 00 00 00 00 00 00 00 00 80 80 80 00   ................
00096  1992 124506  00 00 00 80 80 40 00 00 00 00 00 00 00 00 00 00   .....@..........
00112  1992 125017  00 00 00 00 00 00 00 00 00 00 00 00 80 80 00 00   ................
00128  2018 125529  00 00 80 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00144  1992 126042  00 00 00 00 00 00 00 00 00 00 80 00 00 00 00 80   ................
00160  2024 126555  80 80 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00176  2005 127068  00 00 00 00 00 00 00 80 80 80 00 00 00 80 00 80   ................
00192  2000 127583  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00208  2002 128097  00 00 00 00 80 80 80 80 00 00 00 00 80 80 00 00   ................
00224  2000 128610  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00240  2008 129125  00 00 00 80 80 80 00 00 00 00 80 80 80 00 00 00   ................
00256  2005 129639  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00272  2026 130154  80 40 80 00 80 00 80 80 00 40 00 00 00 00 00 00   .@.......@......
00288  2015 130669  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 80   ................
00304  2007 131185  80 00 80 00 00 80 80 00 00 00 00 00 00 00 00 00   ................
00320  2015 131700  00 00 00 00 00 00 00 00 00 00 00 00 00 80 80 80   ................
00336  2002 132214  00 00 00 00 80 00 00 00 00 00 00 00 00 00 00 00   ................
00352  2015 132728  00 00 00 00 00 00 00 00 00 00 00 00 80 80 00 00   ................
```

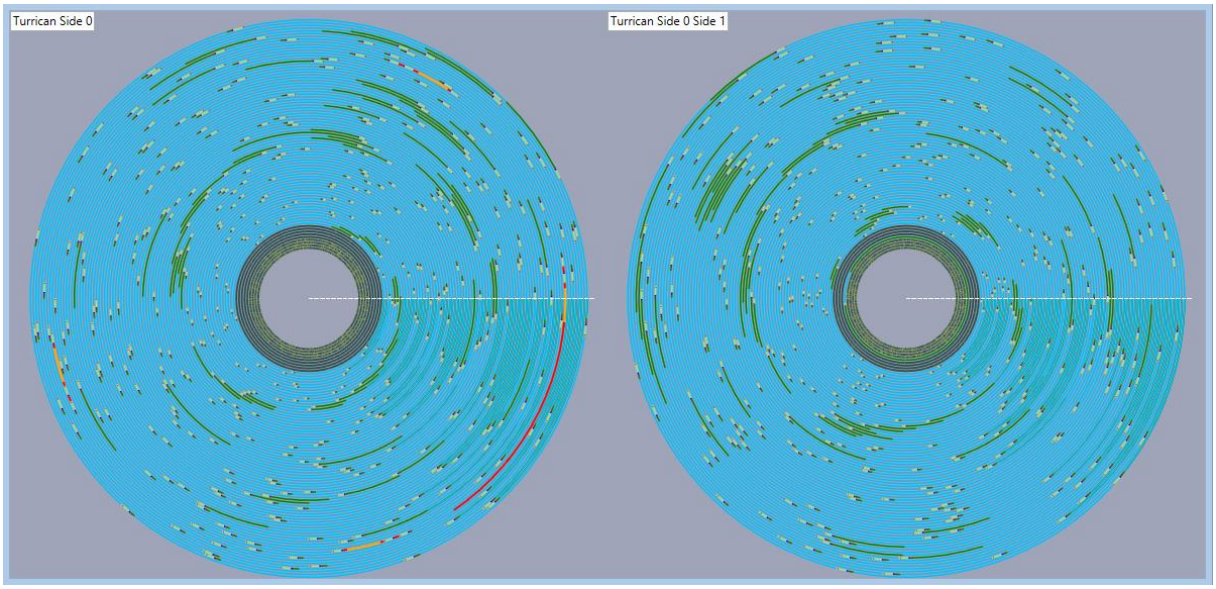
Disk Layout

If you click the **Disk Layout** button a new window will open to display the layout of the two sides of the floppy disk. This display requires to analyze each and every tracks of the displayed side of the Floppy disk. Therefore, unless the track are already analyzed, this operation can take a long time to execute. During this operation the interface do not freeze and it is therefore possible to still select and display different tracks in the main window.

The following color information is used:

- The Sync Character are displayed in Red
- The ID field of a sector is in Blue
- The Intra-gap of a sector is in light green
- The Data field of a sector is displayed:
  - In Deep sky blue by default
  - In blue for over clocked sector
  - In Aqua for under clocked segment
  - In Violet for sector with intra byte speed variation (e.g. speedlock)
- The Inter-gap are displayed in green
- Data sector with CRC error or Fuzzy bytes are displayed in red
- No Flux region are displayed in Yellow

For example the Turrican Game that has many protection and an irregular layout is displayed like this.



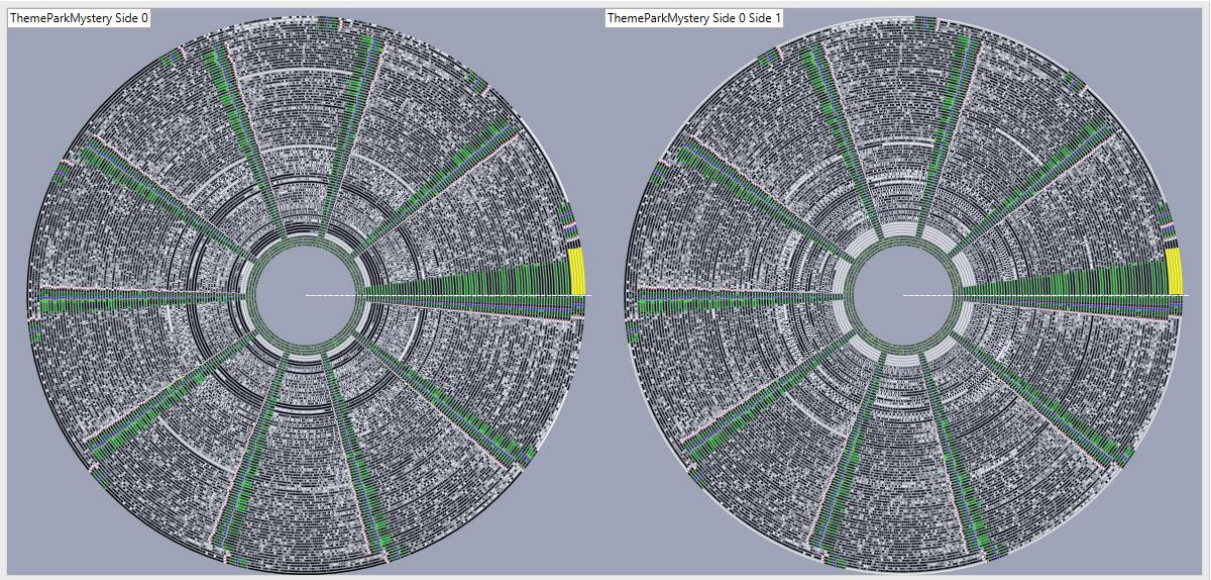
Disk Data

If you click the **Disk Data** button a new window will open and display the data on the floppy. This graph displays a lot of information so be patient.

The following color are used

- The Gap data are displayed using a green color proportional to the data value. For example 0x55 is displayed as RGB={0,55,0}
- The Data uses the same principle but using the three color channels (gray). For example 0x55 is displayed using RGB={55,55,55}
- The ID is displayed in Blue
- The NFA are displayed using yellow color
- The Sync are displayed using Red color

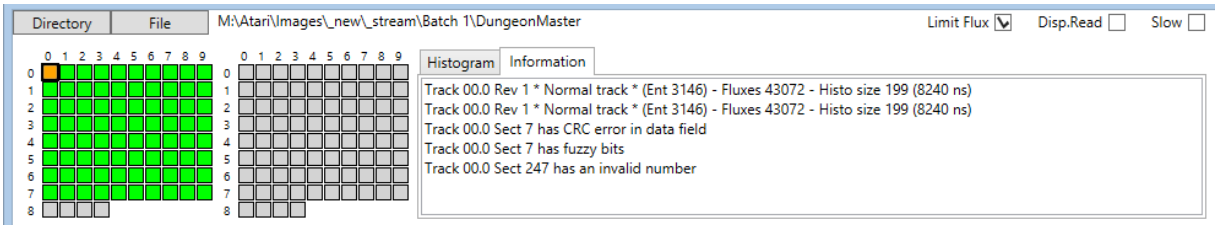
Here is an example of Disk Data Graph on the Theme Park Mystery Game



Protections

Protections are analyzed when a Track is displayed. You can use the **Protection** button to force the analysis of all the tracks. When a track is fully analyzed the background color in the track grid is updated as explained in Selecting the track to display. The **Disk Layout** and **Disk Data** button will also force the analysis of all the tracks. A track that contains protection or error is displayed in Red.

You should select the **Information** tab close to the histogram to display protection information. If you use the **Protection** button all the protections for all the tracks are displayed in the information window.



If you change the currently displayed track by clicking a track in the grid or by using the keyboard direction keys, all the protections for the selected track are displayed.

The following protections are currently analyzed.

- Track Protections
  - Short Track
  - Long Track
  - Track has No Flux Area
  - Track has data over index
  - Track has data beyond index
  - Track has id over index
  - Missing track
  - Extra track
- Sector Protections
  - Sector not in sequence 1..n (used for Pasti)
  - Duplicated Sectors
  - Sector has invalid number
  - Sector has nonstandard size (used for Pasti)
  - Sector has Invalid data into Gap
  - Sector has invalid sync sequence
  - Sector Data has CRC error
  - Sector ID has CRC error
  - Sector has fuzzy bytes
  - Long sector
  - Short sector
  - Intra sector bit width variation (e.g. speedlock)
  - Sector has no data
  - Sector with invalid track number
  - Sector contains sector

For more details about these protections please refer to my document **Atari Floppy Disk Copy Protection**.

Write Disk

This button is used to save the Floppy Disk content using the Pasti STX format.

Beta release limitation:

- Currently the program takes an extremely conservative approach and always save track data and sector information. For more detail please refer to my Pasti File format documentation.
- However track data reuse to save space is implemented.

If the track has not yet been analyzed clicking on the Write track will force analysis of all the tracks.

When you click on the **Write Disk** button a form open and ask you the location and the name of the .stx file to write.

✎ In order to produce correct stx file you must use images with data for at least three revolutions but a value of four is recommended.

If not enough data (revolutions) have been imaged a window will pop up and warn you. You are still offer the capability to force writing the stx file.

If the stx file already exist a window will pop up and ask you if you want to replace the existing file.

Normally single sided / double sided selection is computed by Aufit automatically. If you are getting error messages in Steem that an image of a track is missing on side 1 of the floppy image you may need to force Aufit to write stx file for double sided disk. For that matter please check the ***Force DS*** check Box located in the upper right corner of the window.

Checking Quality of generated images

This section discuss several questions around the quality of Floppy Disk images created with devices like KryoFlux or SuperCard Pro.

To some extent the quality of the image is only related to the floppy drives and floppy disks conditions and therefore independent of the imaging device used.

Cleaning

The first thing you want to do before imaging floppy disks is to clean the floppy drive head and a make sure you have spotlessly clean floppy disk.

🖉 Cleaning the floppy drive head and cleaning the diskette can make a big difference

First always clean the floppy drive head using dedicated cleaning kits as this is a very easy to do operation.

Cleaning a floppy disk is a much more painful operation and therefore you can try to image the diskette without cleaning. If during analysis, as explained below, you find problems on the disk image you should clean the floppy diskette. The recommended way is to wipe the surface of the floppy diskette using a Q-Tip impregnated with head cleaning liquid. Normally this liquid evaporate quickly but you can use the other side of the Q-Tip to help dry the liquid in excess. There are numerous site and videos on the net that explain in detail these operations.

Important things to know when cleaning a floppy diskette:

- The side with an etiquette is the "second logical side" (side/head 1) of the diskette
- The side with the hub that engages with the drive motor is the "first logical side" (side/head 0) of the diskette.

So if you find problems on side/head 0 you need to clean the side without the etiquette (what I refer as the back side) and if you find problems with side/head 1 you need to clean the side with the etiquette (what I refer as the front side).

Analyzing Images

Now let's move on the heart of the subject:

- How do I find if the image I have done is of good enough quality?

For that matter we use the ***Aufit*** program.

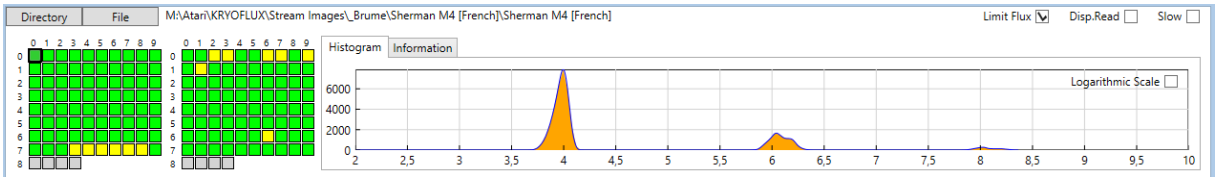
You first need to read the image you have created as explain in Reading Image Files section.

When the program read the image it already display useful information in the Track grids as the background color of each track is an indication of possible problem:

- A "normal" track has a light green background,
- an unformatted track has a light gray, and
- a track with potential problem is displayed in yellow.

At this stage only a quick Shannon Entropy analysis is done on the flux transitions histogram and a yellow color usually indicates track with partially unformatted areas or incorrectly imaged track but always require a more complete analysis.

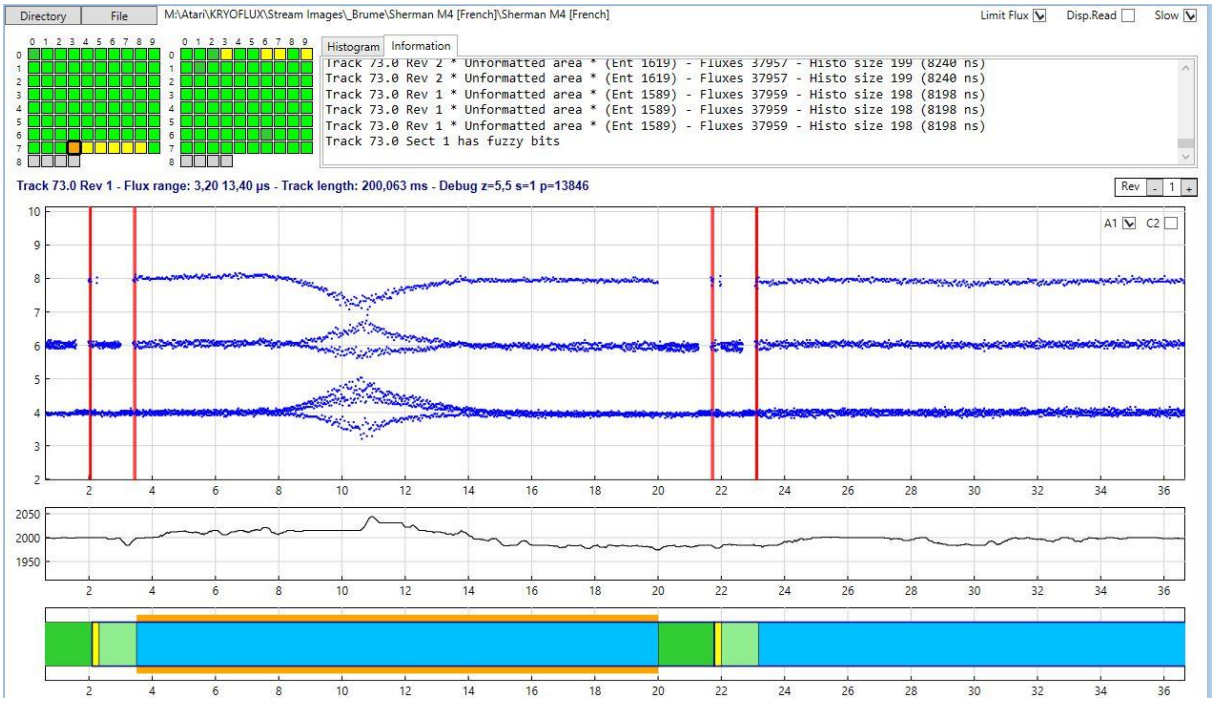
For examples here is the result of reading an image of Sherman M4.



We can see that we have several tracks with a yellow background. If we click on some of these tracks we will find that in fact they are corrects.

But if we click on track 66 we can see than sector one has a lot of "dirty" flux transitions. Thanks to the DPLL these variations can be corrected and we are able to read the sector correctly. This is indicated by the fact that all the sectors read without CRC error.

But if we select the track 73 and zoom on the first sector area we can see that we also have dirty flux transitions, but here the DPLL is not capable to decode the information correctly.



In this case we can see that the first sector is read with fuzzy bytes (orange box in data chart and indication in the information window). In fact if we further analyze the problem we can find, in the fuzzy mask buffer that only **two** bytes at position 206 and 214 vary between revolutions! This also why it is recommended to sample at least four revolutions as **Aufit** always checks fuzzy bytes by comparing the data read for all available revolutions.

```
Fuzzy bytes for sector 1
00000  2000 003505  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00016  2000 004013  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00032  2015 004526  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00048  2015 005040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00064  2005 005554  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00080  2015 006068  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00096  2010 006582  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00112  2018 007097  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00128  2031 007614  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00144  2005 008128  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00160  2005 008644  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00176  2015 009160  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00192  2015 009676  00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ................
00208  2015 010192  00 00 00 00 00 00 01 00 00 00 00 00 00 00 00 00  ................
00224  2020 010709  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

In this case it seems obvious that is error is not part of a protection but due to a problem on the floppy diskette. In such case you should try to clean the diskette and image it again.

We can even look further at the problem. If we click the **Sectors Buffer** command we can see that on some of the revolutions the CRC is good and on some other revolutions the CRC is bad. Normally a fuzzy byte always result in a CRC error. Therefore this indicates that on some revolutions we have read the correct information. Currently Aufit is not able to use this information to create a working Past file but I may implement this feature in the future.

- You need to be aware that on an Atari system when reading sectors from a diskette an automatic retry is performed anytime a sector is not read correctly. This retry is transparent to the user and is used more often than you think.
- Therefore when you have problem clean the floppy and recreate new image, and if this still fails try to recreate the image several times. If the floppy is only marginally not read this may allow you to create a good image.

In order to find all possible problems you first need to run the **Protection** command and for each "protected track" (displayed with a red background in the track grid) you need to check if the problem comes from known protection or from "dirty flux transitions". Unfortunately it seems very difficult to find out about bad images automatically and this is why you need to do a visual inspection using the different features of the Aufit program.

Document History

- V0.4 – March 1, 2014 – Document double sided disk layout, Force DS, and color scheme changes.
- V0.3 – February 17, 2014 – Updated information for release 0.3a
- V0.2 – January 18, 2014 – Added information about usage of the program, about protection analysis, and Writing Pasti files. Added a new complete section about analysis of the quality of the imaged floppy disks.
- V0.1 - January 9, 2014 – Preliminary release for v0.1 Alpha version.
- V0.0 – January 6, 2014 - preliminary first release.

Program History

- V0.4b - March 1, 2014 Color scheme changed, improve disk data, fix pre sync data problem, and fix fuzzy assertion error in compare mode.
- V0.3a February 17, 2014 Check for incomplete information when writing Pasti. Corrected NFA display, improved FDC emulation. More robust: should be able to read corrupted Floppy Disk
- V0.2c January 22, 2014 – Bug fixes on Pasti Files generated
- V0.2a January 18, 2014 – Major change is the capability to write Pasti files. This is still experimental. Bug fixes and better read track report. Now Buffer display uses the currently displayed revolution. This allow to compare the decoded data for the different revolutions.
- V0.1b January 9, 2014 – Fixed one very nasty and tough to find bug. In .Net framework when you click a form to select a file the click event is not flushed and you get it back as a button click ... Also improved memory usage.
- V0.1a January 9, 2014 – Compared to initial release Protection analysis has been added. So only missing initial features is Pasti write capability.
  Known problems: Run out of memory easily, GUI freeze with DISK DATA command.
- V0.0 January 6, 2014 – Initial alpha release on request.