KryoFlux

High Definition Flux Sampler for USB

KryoFlux DiskTool Console (DTC) and KryoFlux firmware engineered and written by István Fábián; additional product design and documentation by Christian Bartsch; Linux port by Adam Nielsen; Mac OS X port by Markus Fritze; DiskTool Console UI by Kieron Wilkinson; hardware layout by Olimex Ltd. & Lars Reichel; Logo by Christian Krapp; project support by the Softpres team

Original hardware design „Cyclone20" and proof of concept by Richard Aplin

Official KryoFlux hardware manufactured exclusively by
OLIMEX

Thanks to our beta testers, especially Dirk Verwiebe. Additional thanks to Jean-François del Nero, Toni Wilen, those we forgot to mention and the communities of English Amiga Board (http://eab.abime.net) and Amiga & Phoenix Community (http://a1k.org) for morale, support and ideas.

Manual Revision 0.97 (2011-09-23)

www.kryoflux.com | forum.kryoflux.com

For commercial use: kryoflux.com/licensing | licensing@kryoflux.com

Usage: Read, convert, store and write contents of various legacy disk formats, including but not limited to: Acorn Electron, Apple, Amstrad CPC, Archimedes, Atari 8-bit, Atari ST, BBC, Commodore 64, Commodore Amiga, MSX, IBM PC, PC-8801, Sam Coupe, Spectrum, E-MU Emulator II and many others.

Preface: KryoFlux by definition refers to the concept of the following project as well as the hardware design itself. While exact details on the hardware and the design are contained in the hardware section below, one should note that KryoFlux is based on the concept of having a small interface adapter using an ATMEL ARM CPU doing the actual sampling. The software side is divided into a dedicated driver for various flavours of Windows, the DiskTool Console (DTC), a flexible software for capturing flux transition data, and the firmware for the ARM board.

Introduction: While todays computers store data on huge hard disks, optical media or even now solid state drives, legacy computers utilized cassettes and floppy disks. Whereas data stored on compact cassettes can be easily sampled using a tape recorder and a sampling device, like a standard sound card found in any modern PC, floppy disks have several shapes and sizes and even more ways to actually store the data on them. Standard PC floppy controllers actually try to interpret and analyse the data before handing it over to the operating system. While some controllers can be tricked into delivering more – raw – data as they should, most of them simply can not be used to read anything but IBM PC compatible formatted media using MFM coding.

KryoFlux replaces any standard controller and makes any attached disk drive available as a serial data stream via USB.

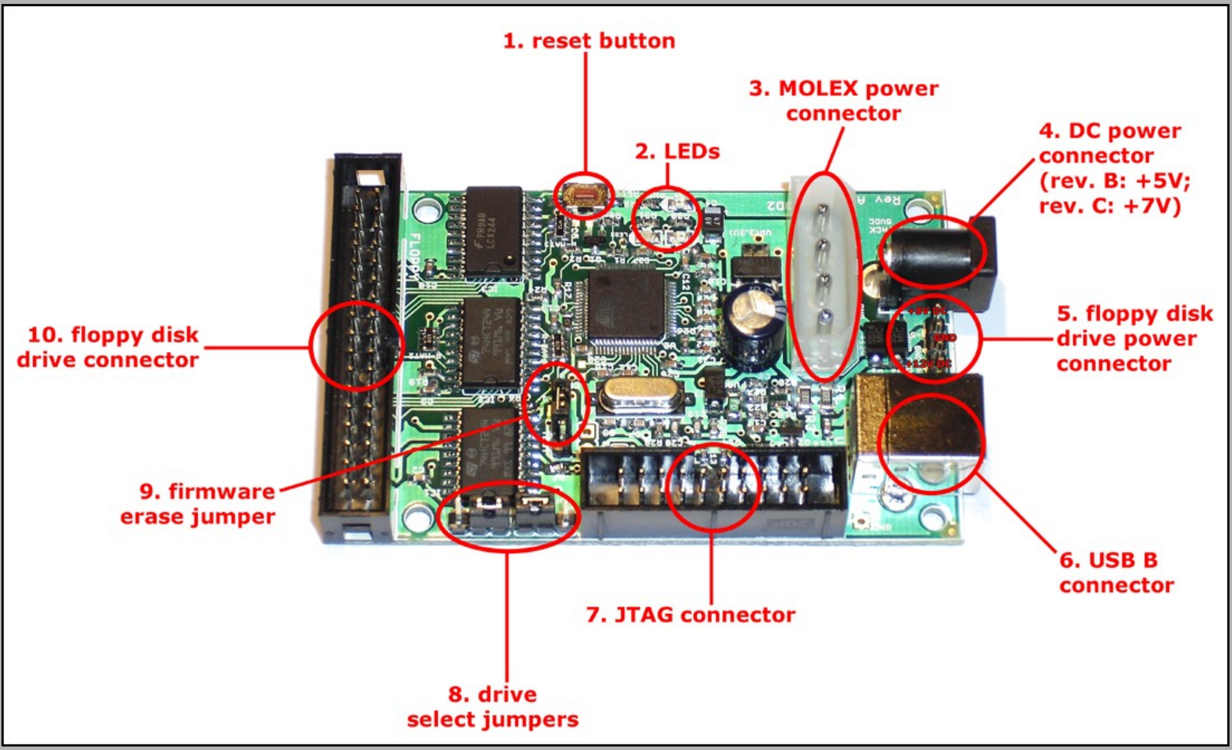
Every magnetic disk, regardless of type or size, stores data by changing the orientation of ferro oxide particles bound onto a durable and flexible plastic platter. The data itself is represented as „flux transitions" aka „flux reversals" which indicate a change of the polarity of the magnetic field. Because it is impossible to actually read the orientation of the particles on the disk surface using the head designs used, the only way to define data is by flux changes. This requires the disk to be spinning because without movement, no AC current is induced in the head. The actual data is normally coded using a scheme like FM, MFM or GCR. While MFM is the most popular scheme (in fact it just survived long enough) used on floppy disks, there are many other ways to encode and represent logical 0 and 1. Error detection and error correction is beyond what is stored in fluxes – both need interpretation of the signal and knowledge about the scheme used for writing to determine if the readout is correct or not.

While optical media produced in a pressing plant can last for ages so to speak, magnetic media has a proven life span somewhere between five to 30 years, with the latter only applying to media which was stored under ideal conditions. The higher the capacity of the platter, the higher the risk of the media failing early.

System Requirements: Computer with 32bit or 64bit flavour Windows (XP or later), Mac OS X (10.5 or later) or Linux; Dual Core, Atom or equivalent processor running at 1.6GHz or more; 1GB of RAM; a native USB 2.0 port; free hard disk space to store tools (~10MB) plus dump data. For best results, KryoFlux must be attached directly to the computer without any hubs or cable extensions inbetween. Due to the precise timing required, results with hubs can be mixed with the possibility of complete failure as well!

You also need a floppy disk drive with a standard 34 pin connector. Please note that KryoFlux was mainly developed for HD 3.5" (e.g. Samsung SFD-321) and compatible HD 5.25" (e.g. Newtronics D509) drives. It also works well with selected 3" (e.g. Amstrad FDI-1) and 8" (e.g. Shugart 851; might require additional adapter) drives. There is a broad range of variants, with some „dinosaurs" not being very keen on standards. It is therefore possible that certain brands or models, especially old drives, may not work with the board. Solutions range from modifying software to modifying hardware. Other types of drives are currently under investigation.

This manual deals with the pre-built and fully assembled board distributed by KryoFlux Products & Services Limited. You can buy the unit directly us via kryoflux.com. *Please note that only units sold by KryoFlux Ltd come with support (as indicated).*



(1) Reset button: If the board does not function or hangs after usage, press this button to reset the board.

(2) LEDs: There are there LEDs on the board. The LED on the upper right (red) should light up all the time when the unit is on. The LEDs to the lower left and right (yellow and green) are off as long the unit has not been used in a session. As soon as the firmware has loaded, the LEDs start to fade alternatingly. The green LED signals firmware activity, while the yellow one indicates an active USB connection.

(3) MOLEX power connector: KryoFlux is a fully bus powered device. Therefore no external power is needed. For special purposes the board allows to be powered externally. It is even possible to distribute power to an attached device (see 5.). Please note that the power rail for +5V is directly connected to the device's CPU. A bad (cheap, unreliable, broken) power supply can damage your board as well as external devices. The PSU must deliver a minimum of 1A per power rail (+5V/+12V). Check the orientation before attaching the plug. *Incorrect orientation of the cable will DESTROY your KryoFlux board and/or your drive. You will also void your warranty (prebuilt boards).*

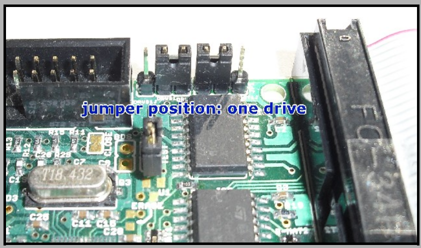
(4) DC power connector: Standard power connector to supply +5V (rev. B board) or +7 to +9V (rev. C board; will internally be transformed to +5V) DC to the board (if desired). Useful when powering a 3.5‟ drive through the board, as these usually don't need +12V. The PSU must deliver a minimum of 1A, tip is hot, shield is ground. *Do not connect more than +5V DC to a rev. B board or more than +9V DC to a rev. C board! You will destroy the board and other equipment as well. You will also void your warranty (prebuilt boards).*

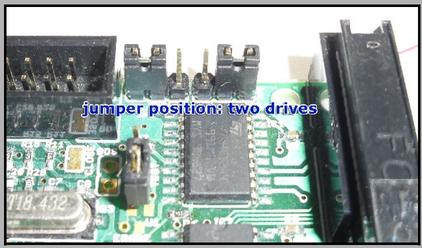
(5) Berg power connector to power a drive – or two with y-cable – via the board: You must use standard Berg connectors (also – incorrectly - referred to as Molex mini) to connect the drive to the board. On the picture above, +12V DC is to the bottom, so the yellow cable of the connector MUST face to the bottom as well, with the red cable facing up (+5V DC). *Incorrect orientation of the cable will DESTROY your KryoFlux board and/or your drive! You will also void your warranty (prebuilt boards).*

(6) USB B connector: KryoFlux connects to the computer with a USB A to USB B plug. The board (not an attached drive!) can be powered solely through USB.

(7) JTAG connector (not used): This connector is for development purposes and advanced servicing only and can be ignored.

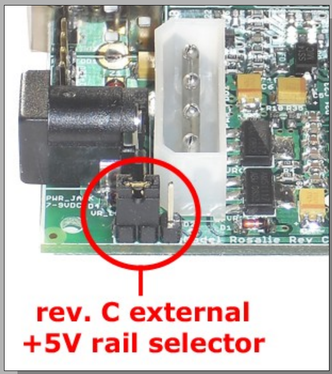
(8) Drive select jumpers: Floppy cables usually have two sections for connecting drives, each of them has two connectors (one for 3.5" drives, the other one for 5.25" drives). You must only connect one drive to one section at a time. The section where the cable is twisted over (at the very end of the cable) is for drive 0 (which used to be drive A: in PCs). The other section is for drive 1 (which used to be drive B: in PCs).



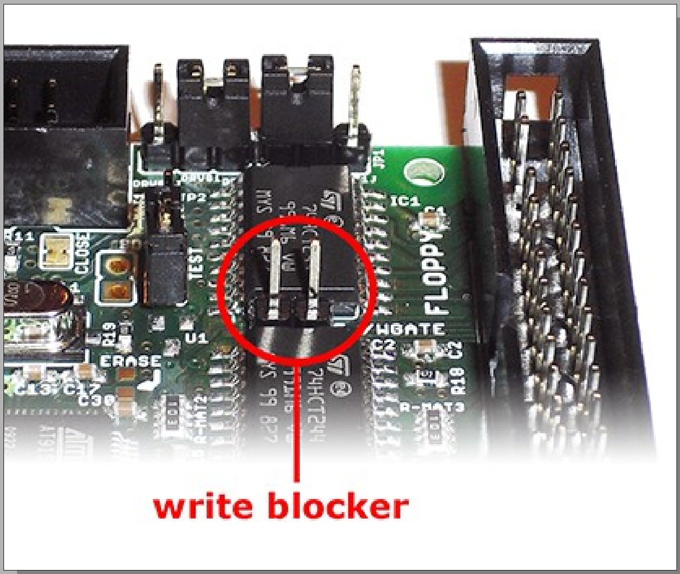


(9) Firmware erase jumper: KryoFlux uses an ATMEL CPU as the core of its system and can be booted from internal flash memory. KryoFlux, as of the time of writing, does not flash firmware onto the device. Instead, it is downloaded at the beginning of each session (it's so fast, you won't even notice). If some other application accidentially writes something into the flash, unplug the device. Set the erase jumper to on. Connect the device, wait at least ten seconds. Now unplug the device and set the jumper to off again. KryoFlux is now back to normal.

(10) Floppy disk drive connector: This socket is for the other end of the drive cable. If it has a small nose, make sure its orientation matches the gap in the socket. If not, please obey line 1, which is marked. Make sure line 1 (usually signalled by a colored cable) is pointing towards the drive select jumpers (no. 8).



Power rail select: Revision C and D of the KryoFlux board come with a regulated external +5V power rail. All external power that is fed into the board via the MOLEX (3) and DC power (4) connector is regulated for a stable +5V DC power supply. This can come in handy for special usage scenarios and will also ensure that the bus driver ICs are operated at precisely +5V. While the DC power connector is always regulated (therefore rev. C boards or later need +7V to +9V DC present at the DC power connector), the routing from the MOLEX connector can be adjusted via a jumper. The setting on the left will transform +12V to precisely +5V. The opposite setting will route +5V or whatever is present at the +5V rail of the MOLEX connector. This rail has no protection diode, so be sure to not experiment with polarity. *More than +5V DC or wrong polarity will fry your board!* We recommend keeping the jumper at the position shown on the picture at all times.



Write blocker: With the introduction of writing to KryoFlux host software (DTC) and firmware, protection of media on the hardware level became necessary. The commercial marketplace offers special devices that can be put between a floppy disk drive and a controller to prevent accidential writes. This functionality has been added to KryoFlux hardware.

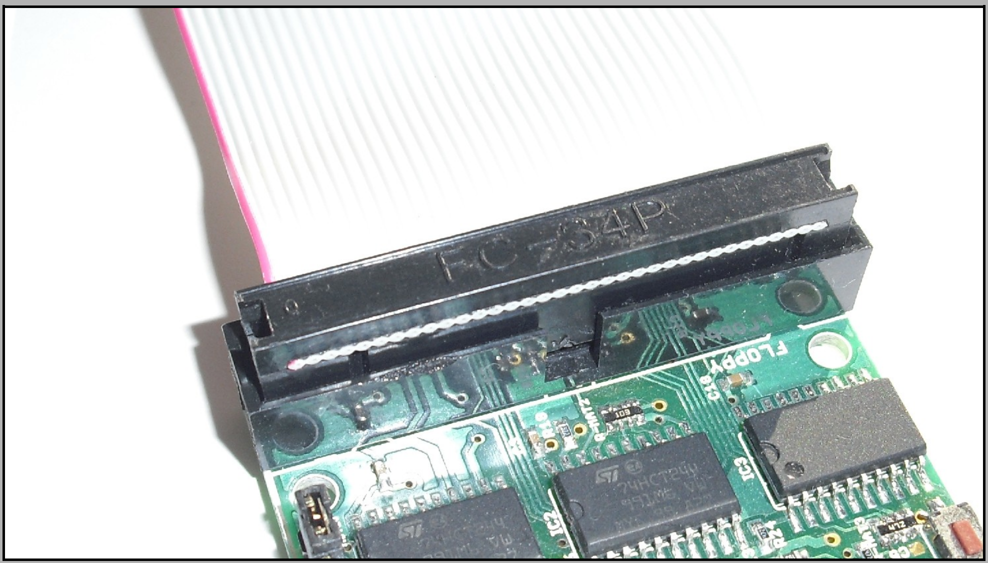
Revision D and later offer a built-in write blocker. The write block can be enabled by removing the jumper for WRITE GATE. After it has been removed KryoFlux can no more write to disk, regardless of media and protection tab. Putting the jumper in place will enable writing again. The picture shows the board with the write block enabled. If you are using KryoFlux in a preservation environment at an archive, library or museum we strongly recommend setting this jumper as shown. This setting can not be circumvented in software.

Setting up the hardware

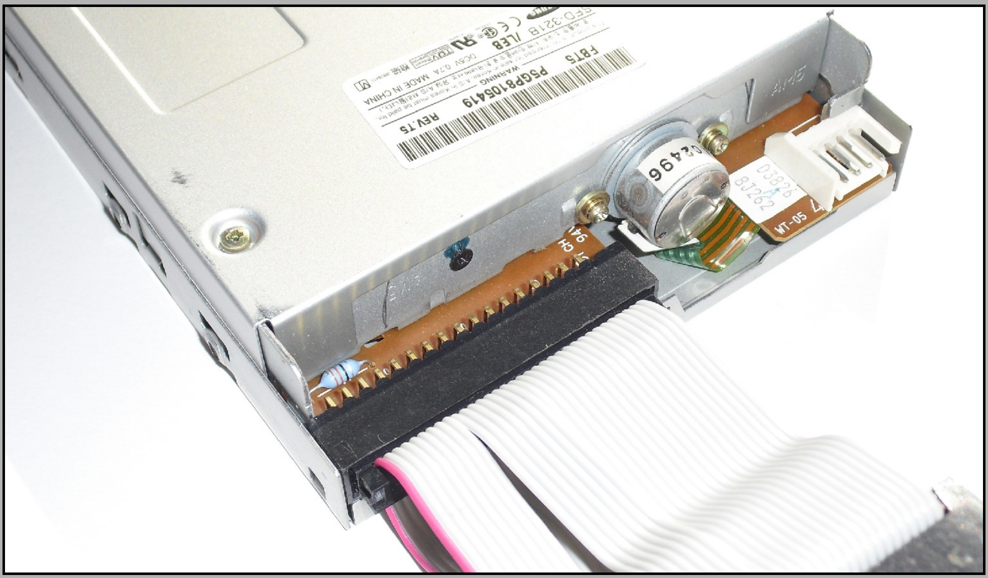
Important! Always make the drive – board - connection first, then plug the power (PSU & USB). Power always comes last! You are connecting two systems with different grounds, so ground (connected via the floppy data cable) must always come first. Never ever connect or remove the floppy data cable while the drive and / or the board are still powered. Doing so will void your warranty and you risk damaging drive and board.

Place the KryoFlux board and the disk drive on a flat, non-conductive surface. Make sure you will not short circuit the device by placing it on a metal table or similar. Connect KryoFlux and drive with floppy data cable.

Always unplug and disconnect from mains when not in use. Do not leave unattended.



Check for correct orientation, the marked wire (usually red or white) signals data line 1. With the board facing towards you and the floppy connector on the upper end, data line 1 is on the left.



Usually, pin 1 must face left when looking at the drive from above with the drive pointing away from you. Still, double check!

Depending on the package, KryoFlux comes with or without cables and a PSU. We recommend powering the drive *directly* with the external PSU. It is possible to route the power through the board, especially, if you happen to have wo drives and only one PSU. You are doing this at your own risk. Please keep in mind that a malfunctioning PSU could destroy your board because of voltage spikes. Connect the drive to the PSU with the Molex plug (if you have a 3.5" drive, you need to attach the Berg adapter to the Molex plug), or the PSU to the board and the board to the drive with a Berg to Berg cable or Berg to Molex cable. Again, we strongly recommend directly powering the drive.

For normal operation, always connect the board to the computer first, then plug in an external power supply. Otherwise you might lock up the board. Simply unplug USB and power, and restart with USB.

Do not power the PSU yet! Do not connect the USB plug to the computer yet!

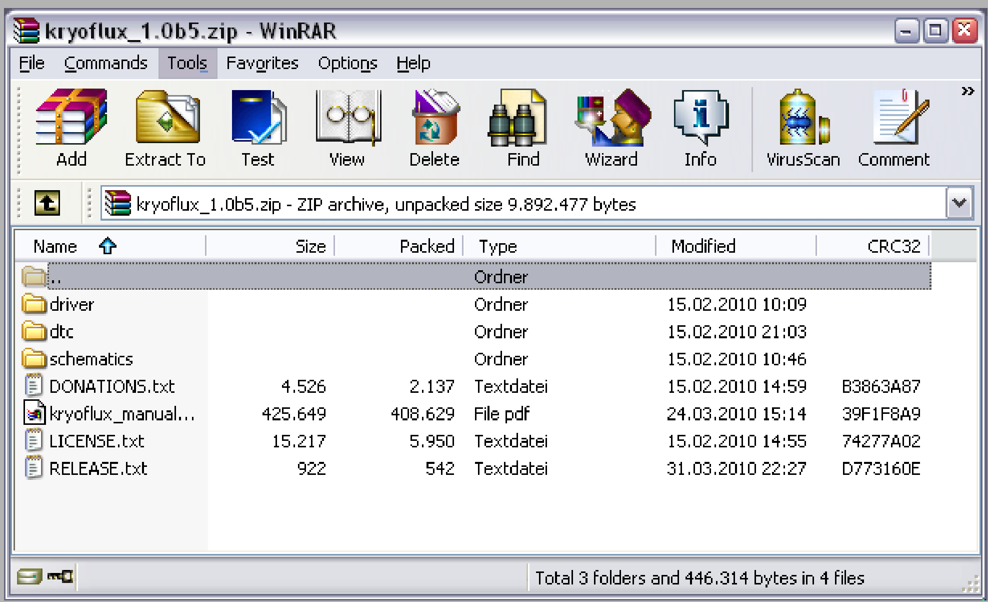
Software Installation

Unpack the software archive available from the KryoFlux web site (http://www.kryoflux.com).

Windows: Copy the DiskTool Console (DTC.exe, firmware.bin, CAPSImg.dll) to a location of your choice. Also take note of the location of the „driver“ folder, as it will be needed to complete the following steps.

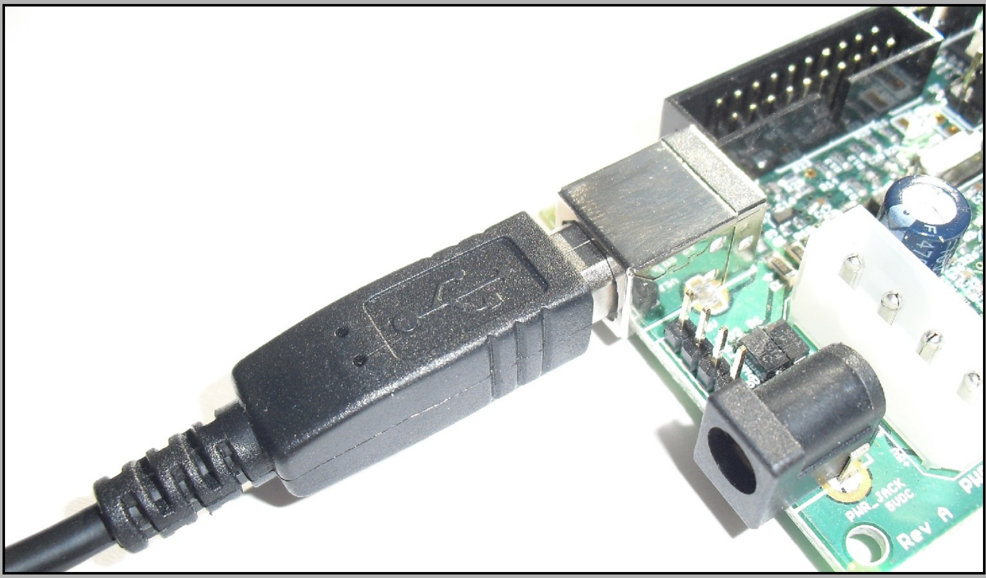
Mac OS X: Just run the installer (KryoFlux.pkg). This will install DTC as well as libusb. The installer includes a text file that contains a list of files and folders installed should you want to remove them later. Please connect the computer and the KryoFlux board with a USB cable (no USB hub!)and continue reading on page 9 („All platforms again“).

Linux: Copy the DiskTool Console (DTC32 or DTC64, firmware.bin) to a location of your choice. Please install libusb 1.0.8 (available separately, chances are it's already installed as this is a popular component). Please connect the computer and the KryoFlux board with a USB cable (no USB hub!) and continue reading on page 9 („All platforms again“).

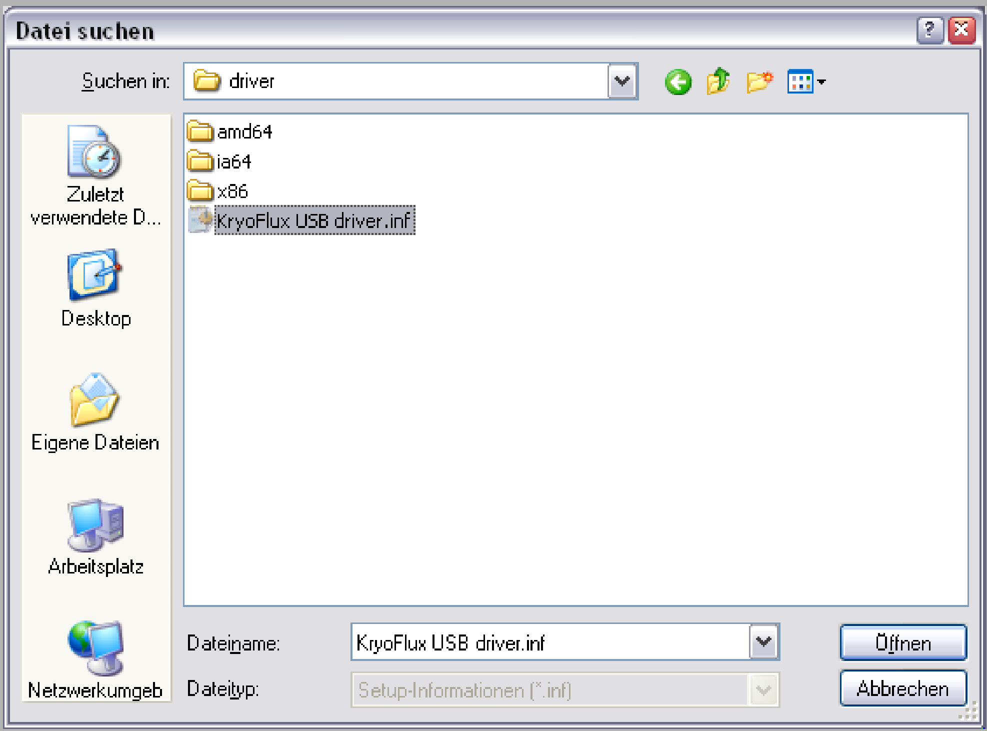


To prevent Windows from searching for a driver for minutes or picking a wrong driver on its own, please disconnect from the Internet for now.

Attach the USB cable to KryoFlux and then attach it to your computer. *Do not use a USB hub!*



The following steps are for Windows only: Windows will ask for a driver, for fastest installation select that you want to pick the driver yourself (usually always the last option in a series of requesters, depending on version of Windows used). Select „KryoFlux USB driver.inf". It is not digitally signed. Wait for installation to finish until you continue.



Open a command line (Start Menu, „Run") and change to the folder where DTC resides. Enter „DTC -c2".

```
C:\WINDOWS\system32\cmd.exe

C:\kf\dtc>dtc -c2_
```

The device will reenumerate, so Windows (versions Vista and before) has to install another instance of the driver. Please follow the same procedure mentioned above. DTC will report an error, which is expected due to the driver being installed.

```
C:\WINDOWS\system32\cmd.exe

C:\kf\dtc>dtc -c2
Timeout while waiting for device initialization

C:\kf\dtc>
```

Plug the PSU into mains. Insert a disk into the drive.

All platforms again: Enter „DTC -c2" (again). DTC will now check for the maximum track your drive can access.

```
C:\WINDOWS\system32\cmd.exe

C:\kf\dtc>dtc -c2
Timeout while waiting for device initialization

C:\kf\dtc>dtc -c2
CM: maxtrack=83

C:\kf\dtc>_
```

KryoFlux is set up. Congratulations! Windows users may now reconnect to the Internet.

Using DTC: DTC is a command line application with an additional user interface that runs on the Java Virtual Machine. The user interface is located in the DTC folder and can be used after necessary drivers have been installed. The GUI will take care of most tasks but currently is not as versatile as the command line version.

DTC is the „DiskTool Console“ and therefore controls all functions of the package. One special feature of DTC is to output several images at once. That means you can e.g. dump an Amiga game disk to a DRAFT raw file while at the same time writing an .ADF of the sector data to see if the disk has a standard file system. You don't have to redump if you find the disk has some kind of protection which can not be represented in a standard sector dump file (e.g. ADF).

You'll find it even more convenient to know that DTC can generate all further images (types 2 and larger) without a disk present. All you ever need to keep are STREAM or DRAFT files. This option is called „deviceless“ mode and means it even works without the KryoFlux hardware present. Think of this as a converter mode, where DTC will operate on a virtual disk, based on a stream dump made earlier.

The only difference between KryoFlux stream files (image file type 0) and KryoFlux DRAFT files (image file type 1 – NOT AVAILABLE IN PUBLIC BETA) is that the latter is a convenient one-file only device-independent image that is handy for transportation, while the former is a group of files representing uncompressed raw data, one for each track and side. This can be more comfortable for development of converters and similar. Please note that generation and usage of DRAFT files should be preferred.

DTC offers the following command line options:

```
-f<name>: set filename
-i<type>: set image type
-m<id>  : set device mode
          1=stream, 2=KryoFlux (default 2)
-d<id>  : select drive (default 0)
-l<mask>: set output level, add values to define mask
          1=device, 2=read, 4=cell, 8=format (default 14)
-r<rev> : set number of revolutions to sample (default by image type)
-t<try> : set number of retries per track, min 1 (default 5)
-a<trk> : set side 0/a track0 physical position (default 0)
-b<trk> : set side 1/b track0 physical position (default 0)
-s<trk> : set start track (default at least 0)
-e<trk> : set end track (default at most 83)
-g<side>: set single sided mode
          0=side 0, 1=side 1, 2=both sides
-z<size>: set sector size
          0=128, 1=256, 2=512, 3=1024 (default 2)
-n<scnt>: set sector count
          0=any, +Z=exactly Z, -Z=at most |Z| (default 0, by image type)
-k<step>: set track distance
          1=80 tracks, 2=40 tracks (default 1)
-v<rpm> : set target system's drive speed, RPM (default by image type)
-x<mode>: set extended cell band search (default by image type)
          0=image specific only, 1=all (default 0 for Emu II, 1 for others)
-y      : set flippy disk mode
-p      : create path
-c<mode>: read calibration mode
          1=track read, 2=maximum track
-w      : write image to disk
```

Image types supported for reading:

```
0 : KryoFlux stream files
2 : CT Raw image, 84 tracks, DS, DD, 300, MFM
3 : FM sector image, 40/80+ tracks, SS/DS, DD/HD, 300, FM
```

```
3a: FM XFD, Atari 8-bit
4 : MFM sector image, 40/80+ tracks, SS/DS, DD/HD, 300, MFM
4a: MFM XFD, Atari 8-bit
5 : AmigaDOS sector image, 80+ tracks, DS, DD/HD, 300, MFM
6 : CBM DOS sector image, 35+ tracks, SS, DD, 300, GCR
7 : Apple DOS 3.2 sector image, 35+ tracks, SS, DD, 300, GCR
8 : Apple DOS 3.3+ sector image, 35+ tracks, SS, DD, 300, GCR
8a: DSK, DOS 3.3 interleave
9 : Apple DOS 400K/800K sector image, 80+ tracks, SS/DS, DD, CLV, GCR
10: Emu sector image, 35+ tracks, SS, DD, 300, FM
11: Emu II sector image, 80+ tracks, DS, DD, 300, FM
12: Amiga DiskSpare sector image, 80+ tracks, DS, DD/HD, 300, MFM
13: DEC RX01 sector image, 77+ tracks, SS, SD, 360, FM
14: DEC RX02 sector image, 77+ tracks, SS, SD/DD, 360, FM/DMMFM
```

Combining of export formats enables stream file dumping with error detection. Therefore, just add the format you want the raw stream verified against as a second export filter. Combined with a large number of retries this sometimes helps rescuing data from worn disks.

Examples:

Remember to set your drive number with option „-d0" or „-d1"!

3.5" AmigaDOS formatted, generate stream files & .ADF file for emulation (e.g. WinUAE):
`dtc -fdumpdir/dumpfile -i0 -fdumpdir/dumpfile.adf -i5`

3.5" AmigaDOS formatted, generate stream files only, do format checks, 500 retries:
`dtc -fdumpdir/dumpfile -i0 -i5 -t500`

3.5" AmigaDOS formatted, only generate .ADF file for emulation (fast!):
`dtc -fdumpdir/dumpfile.adf -i5`

3.5" 720kb DOS formatted, generate stream files & .img file for emulation (e.g. DOSBox):
`dtc -fdumpdir/dumpfile -i0 -fdumpdir/dumpfile.img -z2 -i4`

3.5" 1,44MB DOS formatted, generate stream files & .img file for emulation (e.g. DOSBox):
`dtc -fdumpdir/dumpfile -i0 -fdumpdir/dumpfile.img -z2 -i4`

5.25" 700kb DOS formatted, generate stream files & .img file for emulation (e.g. DOSBox):
`dtc -fdumpdir/dumpfile -i0 -fdumpdir/dumpfile.img -v360 -z2 -i4`

5.25" 1,2MB DOS formatted, generate stream files & .img file for emlation (e.g. DOSBox):
`dtc -fdumpdir/dumpfile -i0 -fdumpdir/dumpfile.img -v360 -z2 -i4`

When writing a sector dump, it is recommended to use option „-l8" to restrict ouput to decode errors only.

IMPORTANT NOTE on command line parameters order: The following settings are "image local" and therefore must appear **before** the image type and would affect only that image type. Their values automatically revert to the default after an image type setting (ie once they get used).

Correct: `DTC.exe -ffilename.ext -v360 -z3 -i4`
Wrong: `DTC.exe -ffilename.ext -i4 -v360 -z3`

```
file name
flippy disk mode
image type
```

```
start track
end track
single sided mode
sector size
sector count
track distance
rpm
```

Other settings are global and can be anywhere in the command line, they'd still affect every operation:

```
device mode
output level
revolutions
retries
track 0 positions
create path *
calibration mode
```

* create path is special as it is only active from the point it's been defined on the command line, ie you can limit which images create their path if needed.

Special, local and global settings:

```
start track
end track
```

If these appear before an image type they affect the image type, then reset to their defaults. If they appear without a further image type setting (that is, anything defined after the last image type command) they'd affect ALL images.

The tracks defined here are primary restrictions; no matter what, DTC will only operate within the global limits. However, when processing each image type, there is a further check if the image has any further constraints and if yes, those can add further limitations to the track range. Once a global setting excluded a track there is no way of adding that back by an image local definition. It could be re-activated by simply limiting the end track, e.g. -e 83 depending on image type

By default images contain all the sides specified by the disk geometry of the image type. If single sided mode is enabled, using an image type that is double sided, but allows single sided image will be forced to contain only the selected side (side 0, side 1 or both). If both sides are selected two images will be created one for each side (naming is automatic with side added) If single sided mode is enabled, using an image type that is single sided, it is possible to select which side will be imaged and disk geometry will be forced to use the selected side(s). Disk geometries for single sided image types only contain side 0, therefore selecting side 1 will transpose the geometry for side 0 to use side 1 instead. In the case of flippy disks (those that were meant to be read with the disk flipped over for side 1) normally a track0 physical position should be defined as well for side 1 (expected to be -8 until further investigation) if the disk is to be read in a single pass without flipping the disk. For dumping disks that may or may not have side 1 formatted, but the target system's drive is capable of reading side 1 without flipping the disk should not have a track 0 position defined. If both sides are selected two images will be created one for each side (naming is automatic with side added).

Flippy disk mode (option -y) reverses the bitstream on the flipside. Note that the position of the index in the bitstream is probably correct only for disks duplicated as "single pass flippy" since those disks used the same index hole for both sides, with modified drives. Disks that were duplicated with earlier drives were actually flipped over, and hence the index is likely to be at a different position. Note that the flipside needs dumping -8 tracks relative to the other side, therefore the drive needs to be able to step to track -8.

Automatic Image Size: DTC automatically creates the output image to be in the minimum size required to represent all sector data without losing content. If a side does not contain valid data according to the image type selected and the format allows it, the sector dump image will be automatically single sided. If tracks do not contain valid data beyond the platform specific minimum number of tracks that should be present in an image, additional tracks won't be added to sector dumps. The minimum number of sectors required to represent all tracks on a disk uniformly (for formats that have this requirement) will be automatically selected.

An image that does not contain any valid data at all in the format selected by the image type will generate a 0 sized file - this is by design. If the automatic image sizing is not desirable (for example an application can only work with a certain number of tracks or sectors) it is possible to change the automatic behaviour by changing the various command parameters - remember those are image local settings and must preceed the image format parameter.

Non-sector dump formats representing low-level disk data (such as stream files and DRAFT) are not affected by automatic image sizing - they always contain all track data dumped.

Sector dump track ordering: Generally two methods are being used to represent track data in most disk image formats.

Option A:
track 0, side 0
track 0, side 1
track 1, side 0
track 1, side 1
[...]

Option B:
track 0, side 0
track 1, side 0
track 2, side 0
[...]
track 0, side 1
track 1, side 1
track 2, side 1

DTC will use common ordering, e.g. A for .ADF and .ST files.

Sectors are always ordered by their physical sector number as stored in the media. The smallest numbered sectors starts at the <track offset>+0 position, the next sector at <track offset>+<sector size> and so on. It does NOT matter whether the system numbers its sectors from 0, 1, 0x41, 0x81, 0xc1 or any other arbitrary value; the lowest sector number found is used as a base for offset 0.

Dump information:

KryoFlux has a very sophisticated cell detection algorithm. Cell analysis is used to identify bits written to disk.

```
base: 2.00 us, band: 4.00 us?, 6.00 us?, 8.00 us?
```

The first value on that line the "base" is the reference clock derived from the type of encoding expected. The following values represent the different bit combinations possible using the encoding scheme (two bands usually used for FM, 3 bands used for GCR and MFM). This interpretation happens depending on the format specified for a sector dump, so trying to dump to two different formats (e.g. .ADF and .D64) would give two lines of results. A question mark indicates that DTC's detection is an estimation only, but it many cases it's still very accurate.

The „-x“ parameter affects which bands will be used for analysation. There are formats that do not fill an entire track, so the rest of the track might contain garbage left over from an earlier formatting. „-x0“ will make sure only bands matching the following format decoder („-i?“) will be used. This does not work for format independent settings („-i0“, „-i1“ and „-i2“).

Writing: DTC can write images onto a floppy disk as well. Due to the way encoding schemes work, only well known data can be written back to disk reliably. Therefore, DTC can write all sector formats supported and IPF files, regardless of format or content. This does not apply to data present in raw images (e.g. STREAM, DRAFT, extended ADF and similar). Although DTC will try to write any image type supported, results for everything that contains raw will be mixed. If you rely on something, using a format that can not be verified should be avoided.

The first public release of DTC only contains write support for IPF files. Other formats are to follow at a later time.

Because DTC does not yet verify data written, it is strongly recommended to only use disks that are known to be good. This can be achieved by formatting the disk on the target platform with verify. It does not matter if tracks are formatted, DTC will unformat tracks as needed.

To write an IPF, please use the following command line:

```
dtc -fimagefile.ipf -w -l15
```

Please note that the framework that handles writing is only preliminary. The syntax is very likely to change. It is not necessary to turn on full logging (-l15), but very helpful to determine what is happening in case you are interested.

Disks used with DTC for writing should be empty. Although you can set start end end cylinders with -s and -e parameters, DTC will erase tracks as needed. DTC will automatically adjust pre-compensation and generate necessary duplicator information. Do not write to a disk with DTC if you don't consider it empty!

Because writing is still experimental (despite the fact it is working very well), disks written with KryoFlux should not be used in a production environment.

Important: Write support requires the latest IPF decoder library (CAPSImg.dll) which is included in the installation archive. If you have installed an older version of it (usually placed in „C:\Windows\System32“), overwrite it with the one supplied. It is recommended to store the library in one location only.

About the Software Preservation Society (SPS)

SPS is a privately funded association of art collectors and computer enthusiasts striving for the preservation of computer art, namely computer games.

Art is an important cultural asset. Thousands of museums and archives all over the world preserve and restore pictures, books, movies and audio recordings and information in general for generations to come. To accomplish their assignment, national libraries are backed by law which, varying from country to country, forces production companies to deliver copies of publications, books, audio recordings and movies to the archives for long term preservation. It seems that as of today, nobody has ever thought or actively cared about the true, unmodified and verified preservation of computer games. Without any action taken, time will run out, very quickly.

Unlike games from the 1970s (delivered on solid state ROM-modules) and games from and after the mid-1990s (delivered on optical media like CD-ROMs and DVDs which are supposed to last for decades), computer games from the 1980s and early 1990s were delivered on magnetic media like tapes or floppy disks and are now at the brink of extinction.

From a preservation point of view, tapes and floppy disks are a nightmare for several reasons:

1. Tapes and floppy disks constantly degrade, in two ways. First is the physical degradation of the orientation of the metal particles which form the magnetic field and store the data. This process is slow, and given the fact that the data is encoded digitally, it may be too late to do anything when reading errors occur. Reading errors happen when it has become difficult to decide if a particular bit is 0 or 1. Preservation should occur before it becomes a gamble to get a good read.

2. Second is the chemical degradation. The metal particles bound to the plastic platter of a floppy disk or the surface of a tape can come off the surface. In fact, in most cases the bonding will simply fall apart after years of temperature changes, moisture and other issues of improper storage. Record companies struggle with this problem when remastering old recordings and have developed a process called baking where the original master tape is actually put in an oven to rebind the coating to the transport material. After baking, playback is a one try only process because the media will fall apart after passing the playback head of the machine. While similar to the original is sufficient for analogue material, even a single misinterpreted bit in the digital world means instant failure.

3. While no user can actually press industry standard vinyl recordings, CDs or DVDs at home (recordable media can be spotted by simply looking at it), tapes and floppies can actually be written and modified with consumer-grade equipment. It takes a lot of expertise to distinguish a professionally replicated medium from a home made copy. Even if a disk was produced by a commercial replicator, it does not necessarily mean that disk is still authentic and appropriate for preservation. Apart from a game possibly being copied over the original (as we have seen many times to „fix“ a broken disk), many games themselves persist some kind of save state or high score, thus changing or erasing data that was available on the disk in the first place. As soon as the disk has been modified in any way, the authenticity of that copy is put into serious doubt.

SPS has successfully mastered these challenges and developed software and hardware technology to deal with the problems arising during the preservation process. Founded by computer expert and preservation pioneer István Fábián in 2001 as CAPS (the Classic Amiga Preservation Society), our highly specialized team has more than nine years of field experience. SPS members have not only been involved in playing games on the machines which are regarded retro today, but were programmers and designers also responsible for some of the games and programs available on these platforms.

While our original disk imaging tools (working on e.g. a standard Amiga 1200 with a compact flash adapter) are still good and easy to use, we are currently moving on to a completely self-contained floppy controller „KryoFlux‟ developed by SPS that works with any modern PC via an USB connection. This does not only speed up imaging of disks, but also enables physical media restoration of any title preserved so far.

Preservation at SPS usually is a two step process. Contributors from all over the world can help imaging disks with our unique technology. At SPS, our experts then use the Softpres Analyser to investigate the disk structure and create an IPF (Interchangeable Preservation Format) file. Scripting allows a flexible, even game-specific, way of representing data when read by a tool, or when rewritten to disk. Often rather different methods are required to represent various disk formats or copy protection methods when intended to be read by e.g. an emulator or to be written back when restoring an original disk. Due to the high quality of the preservation technology, IPFs have become the de facto standard demanded by Amiga users when looking for unmodified images true to the original.

While disks themselves are the problem that needs to be addressed quickly while they are still readable, SPS is also striving for complete archival of manuals and boxes in the form of physical products as well as digital scans. As of today, SPS has digitally archived about 3.700 games produced for the Commodore Amiga, Atari ST and others. Complete support for other platforms, like the C64 (which is a real challenge due to a second „computer‟ built straight into the floppy drive) is in the works, but disk imaging of such material already works today. It is only a question of manpower when the data imaged will be ready for presentation in dedicated IPF files. Again, this is a race against time to protect gems of yesterday from fading into oblivion.

For more information: softpres.org

Contact: softpres.org/contact