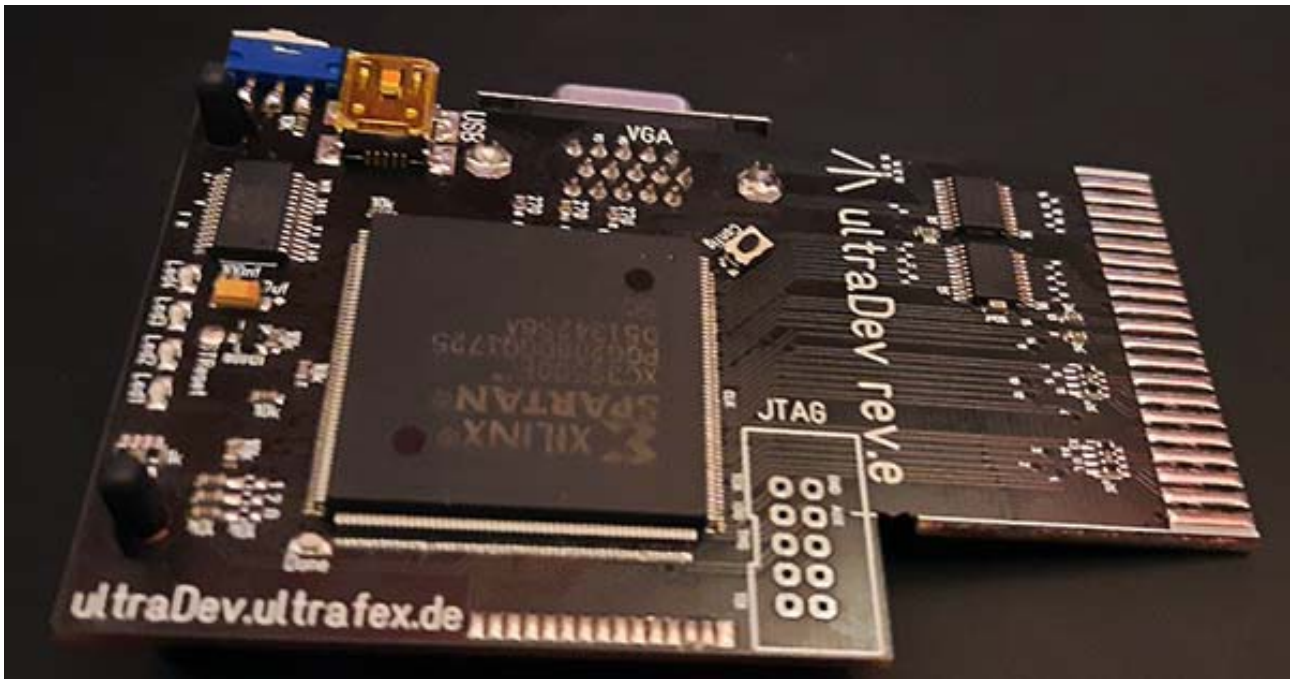
ultraDev

The Atari Development Cartridge



User Manual

Version 0.56

This page intentionally left blank

I always wanted to write that ;)

Table of Contents

1 Welcome

ultraDev is a FPGA based development cartridge for Atari computers.

How does it works?
After a reset the Atari stays in an upload screen and waits for an upload from the host computer. You can upload and start a Prg-file from the host computer command line via USB.

With a small cable from inside the Atari the cartridge is able to reset the Atari.

Which means you just run the command line tool again and the cartridge resets the Atari, uploads and starts the prg.

If you do not want to open up your Atari and solder in the cable you can also use a small resetter code which is included in the 68k library.

For more informations check Working with(out) a reset cable

How about debugging?
If you want to debug you can tell the command line tool to start bugaboo before and directly load the Prg-file.
With the command line tool it is possible execute bugaboo commands after bugaboo loaded the Prg-file. Very useful to set breakpoints or just execute automatically after loading.

It's also possible when using the keyboard mode to send key strokes from the pc command line window directly to Bugaboo.

Debug screen
You can connect a VGA monitor to the cartridge to output debug text values on a separate screen (40x32). You can use up to 8 colours. It is even possible to upload a font to create custom chars. A bitmap mode with 128x128 resolution is also supported. Have a look to the 68k Library and examples how to use it.

1.1 Electrostatic-sensitive device warning

ultraDev is an electrostatic-sensitive device. What does this mean? (Text from Wikipedia)

"An electrostatic-sensitive device (often abbreviated ESD) is any component (primarily electrical) which can be damaged by common static charges which build up on people, tools, and other non-conductors or semiconductors. ESD commonly also stands for electrostatic discharge.

As electronic parts like computer central processing units (CPUs) become packed more and more densely with transistors the transistors shrink and become more and more vulnerable to ESD."

ultraDev has no case if you touch the PCB you can damage the components. So it's a good idea to touch a grounded device before you touch the PCB.

But it's not that dramatic how it sounds. Often I forget to touch a grounded device before and till now never something happened. **But you have been warned :)**

1.2 Open source

If the project is in a final state the sources will be released. Currently only the firmware for the 68k firmware source and the modified Bugaboo source is available in the release.

2 Included parts

Delivery of ultraDev includes: The cartridge, 3 connectors for the reset cable and 3 shrink tubing and 2 spacer bolts.

A USB-b cable is not included.

3 Support and Bugs

If you have problems, questions or you found bugs you can reach me at: ultrafex1@gmx.de

I'm also quite often on irc (ircnet) channel #atariscne.

Feature suggestions are also welcome. I'll decide then if I will implement them.

4 ultraDev Software

Releases can be found at: http://ultradev.ultrafex.de/releases/

5 Setup & Install

5.1 Requirements

- As host computer only Windows (XP-W10), Raspberry PI (32bit only) and Mac (64bit only) is supported.
- The cartridge should work on:
  - ST(E)
  - Mega ST(E)
  - Falcon. CT60 is currently not tested
  - TT

5.2 Release folder structure and files

- 68kFirmware

  I decided the Atari 68k sources for the firmware is in the release. Maybe

better if you have problems with starting something you can have a look to the sources.

Also the preassembled firmware.img you can find here. Normally you don't need this. The firmware.img is included in the FPGA firmware.

- 68kLib

  Library to handle the cartridge like printing to screen and stuff. There is a more detailed description a bit below (68k Library)

- 68kLibSamples

  Examples how to use the 68klib there is a more detailed description a bit below (68k Library Examples)

- Cmdline

  Windows Command line tool to upload a prg to the Atari (Uploading and starting a Prg-File).

  Raspberry Pi commandline is placed in linux.rpi (32bit only)

  Mac commandline is placed in linux.mac (64bit only)

  The special Bugaboo version is included here.

- Doc

  Usermanual.

- FPGAUpdate

  A small script which calls ultraDev.exe to update the FPGA firmware. How to do this see (Updating the FPGA firmware)

5.3 Driver install

5.3.1 Windows

Go to the Driver directory and execute CDM21228_Setup.exe. Follow the setup instructions.

After Install the USB device „ultraDev" should show up in the "Devices and Printers" panel in Windows.

You can connect ultraDev to one of your USB ports without connecting it to the Atari. Just to see if it's showing up there. Of course upload do not work then...

5.3.2 Raspberry Pi

Keep in mind currently the command line tool is a 32bit exe only !

Download the drivers at https://www.ftdichip.com/Drivers/D2XX.htm

Linux →1.x.x ARMv7 hard-float ***

On the left in the Linux line you can see a ReadMe how to install.

Follow the instructions. Only "Installing the D2XX shared library and static library" is

needed.

Then you should be able to start ultraDev placed in Cmdline\linux.rpi\

You need to start the it with sudo ultraDev !

On Raspberry there is an sio driver for the USB chip available which do not work with the ultraDev libraries.

When starting ultraDev.exe it executes by default:

sudo rmmod ftdi_sio

sudo rmmod usbserial

5.3.3 Mac OSX

Keep in mind the command line tool is a intel 64bit exe and works for OSX 10.4 or later only

Download the drivers at https://www.ftdichip.com/Drivers/D2XX.htm

Mac OS X 10.4 Tiger or later →1.x.x

On the left in the Mac OS line you can see a ReadMe how to install.

Follow the instructions. Only "Installing" is needed.

Then you should be able to start ultraDev placed in Cmdline\linux.mac\

You need to start the it with sudo ultraDev !

6 Hardware install

First plug in the spacer bolts into the holes at the end of the PCB! This ensures the correct placement of the PCB when the cartridge is connected to the Atari.

Installing the cartridge to your Atari is pretty easy. Just plug it in ;)

The cartridges has a switch at the end of the PCB. Moving the switch away from the Atari the cartridge is switched on.

If the cartridge is swiched on you can upload Prg-Files with the command line tool. In this mode it's not possible to use the HD Emulation.

To use the HD Emulation move the switch to the Atari.

Do not wonder if the cartridge screen does not show up directly after booting. The FPGA needs some time to boot so it misses maybe check from the TOS if there is a cartridge. If the Done Led lights up (on FPGA board) you can press reset and if the cartridge is switched on the cartridge screen will appear.

Installing the Atari reset cable is currently not documented. If you wish the Atari resets each time before a new Prg-file is uploaded have a look to the 68k Library Samples. There is a small inserter code which resets the Atari if needed.

When not using the reset cable you should leave the small sticker over

the connector (which i stick over when I send it). That connector has 5v if this connects to your FPGA it will damage it !!!

6.1 Connecting a VGA monitor to the Cartridge

Under the cartridge you will find the connector for the VGA screen. ultraDev creates a 1280x1024 @60hz signal.

Have a look to the 68Lib and 68LibExamples how to write stuff to the screen.

6.2 Leds, buttons and switches

Following Leds are on the FPGA board:

- Power Led

  Lights up if the FPGA got it's power

- Done Led

  After power up the Led is off and will light up when the FPGA loaded the Firmware from the flash. If it does not light up you have a real problem. This will mean you need a XiLink programmer to fix that.

- Led1-4

  Led 4 blinks slowly if the cartridge is in idle and waiting for data from the host computer. If no USB cable is connected it blinks a lot slower

  Led 3 blinks slowly if the cartridge waits for data during a command. If it is blinking and the commandline tool hangs there went something wrong. To

  fix that you can press the config button. This will restart the FPGA.

  Led 2 shortly lights up when a sector is read from the host computer.

  Led 1-4 blinking fast the cartridge waits for the Atari to be resetted.
  If you have a reset cable installed the cartridge recognises if the Atari preformed a reset and continues

  If you don't have a reset cable installed you need to press reset by yourself.

  If the FPGA is in that blinking mode no other upload is accepted. To get out of this state press reset on Atari or use the config button.

  By the way there is a resetter inserter code for your VBL routine which does the reset for you have a look to the 68kLib and 68kLibExamples how to use it.

All buttons are on the FPGA board:

- Config will restart the FPGA. If you have a strange behavoir like the cartridge do not react this is the button to press

Switch on the cartridge:

- This will switch off or on the cartridge. If you do not want the cartridge boots into the ultraDev screen you can switch it off. Only if the cartridge is off you can use the HD Emulation.

Move the switch away from Atari to switch the cartridge on.

7 Atari memory used by ultraDev

ultraDev uses about 60 kb of the Atari memory when it's switched on. You can see the available memory on the cartridge screen.

If switched off and no folder share is active it will not need additional memory. If a folder share is active the cartridge will allocate additional memory for the drive caches amount depends on the sector size.

8 ultraDev.exe command line tool

8.1 Command line parameters

ultraDev.exe is placed in the directory "Cmdline".

- -prg (path) Start Prg-file.
- -dbg (path) Bugaboo path. If given bugaboo is started before.
- -cmd (Bugaboo commands) Bugaboo commands which are executed after loading. -cmd g starts the prg directly after loading. The commands are separated by :
- -uc (path) Upload cartridge (68k firmware). Normally you don't need to upload a 68k firmware. The current firmware is included in the FPGA. This is just for testing reasons or if you want to change something to the 68k firmware.
- -data (path) Send data of max 16kb to the cartridge. This can be used to send data after a prg is started. The 16kb is visible from $fa8000-$fabfff.
- -fu (bit-filepath) FPGA update. Write the bit-file content to the flash config rom. All other parameters are ignored then!
- -share (path) Shares a PC folder to your Atari. See "Sharing a PC folder to your Atari (HD Emulation)" for more informations.
- -reset Force a reset on Atari
- -driveletter (c-n) Driveletter for -share. Default is c.
- -sector (512/1024/2048/4096/8192) Sets the sector size for the volume 8192 is default.
- -hddriver HDDriver compatiblity sets sector size to 4096. -hddriver wins over -sector
- -key (char/0x<hexvalue>) Send a char to the Atari which can be with UDKeyStroke. **(Windows only)**
- -kbmode Starts the keyboard mode. ultraDev does not exit and sends every key to the Atari. Can be used during debugging with Bugaboo to send key strokes from the pc keyboard directly to the Bugaboo screen. **(Windows only)**

- -serial <serialstring> When using multiple cartridges on a host computer you can use -serial to use a specific ultraDev cartridge
- listdevices Lists all serial strings of ultraDev cartridges connected to the host computer

8.2 Uploading and starting a Prg-File

If you want to upload a prg to your Atari your cartridge needs to be switched on.

With: ultraDev.exe -prg <path>

The prg file is automatically started. If a reset cable is installed the Atari resets just before the prg file is started otherwise you need to press reset by yourself.

You can also just drag and drop a prg file to ultraDev.exe to start a program.

8.3 Debugging with Bugaboo

When debugging with Bugaboo it's important to use the Bugaboo version included in the release! Otherwise prg loading will not work.

8.3.1 Changes to Bugaboo

The reset saveness has been removed from Bugaboo. This is needed otherwise the reupload of a new Prg-File don't work.

The le (load executeable) was changed only to load from the cartridge. So do not use ! The le command is always added to command list after loading bugaboo to load the Prg-File.

8.3.2 Starting a debug session

ultraDev -dbg <path to the ultradev Bugaboo version> -prg <path to the prg file>

Starts a debug session. Bugaboo and prg is loaded. If the prg file has symbols you can of couse use them to debug.

By default the programm is not started automatically. If you want to start the prg directly after loading do it with:

-cmd g

Multiple commands must be separated by a :

The commands are executed directly after loading the prg file.

8.3.3 Using the pc keyboard to type during a Bugaboo session

With -kbmode you can start the keyboard mode. ultraDev will not exit and sends all key strokes from the pc cmd window to the Bugaboo screen.

The keyboard mode does not display cursor movements.

This feature is currently Windows only.

8.4 Using multiple ultraDev cartridges with the host computer

All ultraDev cartridges have a unique serial number. If you are using more than one cartridge with the host computer you can choose with -serial <serialstring> the cartridge to use.

So see the serial number for the ultraDev cartridges you can use -listdevices. This lists all connected ultraDev cartridges including the serial string.

8.4.1 Changing the serial string of a cartridge

The serial string is auto generated during the first program of the usb chip. It's normally quite unreadable. To have a more readable serial string you can change it.

ultraDev uses a usb chip from FTDI. There is a tool call FT_PROG which is able to change the configuration of the usb chip.

Download the tool from the FTDI site.

I recommand to disconnect all ultraDev cartridges except the one you want to change the serial string. Also disconnect a usb floppy emulator it uses the same chip.

Start the FT_PROG, select from the menu "Scan and Parse" from the menu.

The cartridge should show up now. If you select in the Device Tree "FT_EEPROM" you should see on the left under "Product Desc" ultraDev.

Select in Device Tree "USB String Descriptors" on the left you can see "Auto Generate Serial" switch that off. Directly below you are now able to set a Serial Number. Set it for example to "Atari1040STE". Avoid spaces here!

Select from the devices menu "Program Device" and click Program.

Disconnect the cartridge again and wait some seconds and connect again and you are done.

8.5 Sending data to the cartridge

With -data <path to the data> it's possible to send data to the cartridge (max.16kb) the data appears at address $FA8000. This is only possible if no folder share is active.

8.6 Creating own cartridges

It is possible to create own cartridges which stay resident until switch off of the Atari.

With – uc <path> you can create your own cartridge (max 16kb).

Important is that the cartridge needs a special header otherwise the cartridge is not recognised by Tos. Have a look to main.s there is a working header included.

Important 2 the first instruction in your code should be:

```
move readyFlag,d0        ;tell the FPGA the reset is done
```

this is to tell the FPGA the reset is done. If you don't do and your initiate a reset with the command line tool all cartridge leds will blink forever.

8.6.1 Creating own cartridges with 32kb

Creating cartridges with 32kb is only possible if the cartridge space is segmented means:

Segment 1: $fa0000-$fa4000

Segment 2: $fa8000-$fac000

To upload the code do: -uc <path> -data <path>

-uc is for segment 1 -data for segment 2

9 Sharing a PC folder to your Atari (HD Emulation)

The command line tool ultraDev is able to share a PC folder to your Atari.

Your cartridge needs to be switched off.

The HD emulation has no write support !

Changes to the folder are recognised by ultraDev and after the folder is stableyou can use the file on your Atari. **(Currently only supported on Windows!)**

With the command line parameter -share <path> ultraDev.exe starts up in a server mode and with the next reset of your Atari a volume C is installed to the desktop.

The volume is only installed when the ultraDev server is running. By default the drive letter is C if you want to set a different you can use -driveletter (c-n)

In the server mode ultraDev.exe keeps running and serves the requests from the Atari.

If you place a DESKTOP.INF on your share root folder the Atari uses it.

To quick start a folder share session you can drag and drop a folder to the ultraDev.exe.

On Windows you can also use network volume (unc pathes).

By default (8k sectors) the HD emulation is able to create 512mb volumes. The root directory can hold 512 files. Sub directories are limited to 1024.

The transfer rate on my 1040STE with my Windows host computer setup is about 820kb/s. On tin's falcon it's about 900kb/s.

Rpi Hd emulation is alot slower with my Rpi 3 I have about 450kb/s.

9.1 Compatiblity with HDDriver

By default a folder share uses 8kb sectors to maximize speed and volume size (max 512mb). When using other volumes with HDDriver you may need to change the sector size by calling ultraDev.exe with -hddriver.

This sets the sector size to 4k without the reduced sector size hddriver volumes will not work.

With -sector (512/1024/2048/4096/8192) it's possible to set other sector sizes but this will of course reduce your volume size.

10 Updating the FPGA firmware

New firmware releases can be found at: http://ultradev.ultrafex.de/releases/

10.1 Updating with the FPGAUpdate.bat

To update the FPGA firmware you need to plug in the cartridge into your Atari, power up, connect the USB cable to your host computer. If the cartridge is switched on or off makes no difference.

On Windows:

Double click FPGAUpdate.bat placed in the FPGAUpdate folder. Follow the instruction on screen. **DO NOT SWITCH OFF THE ATARI DURING update !**

On Linux/Mac:

ultraDev -fu <path to top.bit>

Follow the instruction on screen. **DO NOT SWITCH OFF THE ATARI DURING update !**

The update needs about 15 seconds.

When the update is done switch off/on your Atari or press the config button on the cartridge and preform a reset on Atari. After pressing the config button the done led should light up.

Best way to see if the update worked fine is to switch on the cartridge. On the cartridge screen on the right side you can see the version number.

Keep in mind that after an update you need to use the same version number of the command line tool. Otherwise you will get an version mismatch error message.

If you recognise a problem with the new version you can down grade to an older version this is working without problems

11 Working with(out) a reset cable

When uploading Prg-files during development the best way of using ultraDev is when you install a reset cable.

The reset cable is just a small cable which is soldered to a IC. Even if you are not a soldering king this is possible to do. When you upload a Prg-File the Atari is resetted automatically if needed.

If you plan to add a reset cable contact me i'll help you to install it.

It's of course possible to work without a reset cable. When you upload a Prg-file and a reset needs to be preformed all leds on the cartridge will blink.

If the cartridge is in this state you must do a reset by yourself with the Atari reset button means the cartridge will only leave this state if the cartridge recognises a reset from the Atari.

If the cartridge in this state you could also leave this state by pressing the config button on the cartridge. This will reset the FPGA on the cartridge.

Have a look to the 68k library there is a small source UDResetter.s which can help you working without a reset cable.

Include UDResetter.s to your source and add a jsr UDResetCheck to your vbl routine. If you upload a new version of your Prg-File via the command line tool it resets the Atari automatically to load it.

If your code crashes this will not work you need to do a reset by yourself. Another idea is to install exception handlers which will reset the Atari automatically. A reset you can do with:

```
lea     $4.w,a0
move.l  (a0),a0
jmp     (a0)
```

12 68k Library

ultraDev comes with a small library which helps you to use the cartridge features without coding the new hardware directly. The 68k library is located in the release 68kLib.

There are a few examples how to use the 68k Library check 68k Library Examples.

Have a look to the sources which routines are available I tried to write some more informations how to use and stuff.

Files included:

- UDAll.s

Includes all ultraDev includes

- UDBitmap.s (VGA screen must be connected)

  Service routines for the bitmap mode. The bitmap mode displays a 128x128 image. The resolution is halved to fill the screen.

  How does it work? More or less is the bitmap mode very like on vc20. You just print the chars from 0 - 255 on screen and the bitmap itself is modified by changing the font.

- UDCartridge.s

  Service routines how to detect the cartridge clean and stuff like version numbers etc.

- UDDefines.s

  Includes defines for the new hardware registers

- UDFont.s (VGA screen must be connected)

  Service routines for uploading a font.

  Keep in mind if you upload a new font into the FPGA this font will remain in the FPGA until you restart it. Means if you fucked up the font it will stay fucked up. But there is a routine which reinitialises the font if you want

- UDResetter.s

  Small inserter code to your vbl routine. If you upload a new version of your Prg-File via the command line tool it resets the Atari to load it. Only needed if you do not have a reset cable installed.

- UDTerminal.s (VGA screen must be connected)

  Can be used to create a terminal like screen area which means you print text and at the end of the screen the text scrolls up. By setting the start line for example to 16 you can have a scroll area at the bottom of the screen upper area is not touched means you can show stuff here...

  use UDTermPrint to print stuff into the terminal.

- UDVideo.s (VGA screen must be connected)

  Service routines for the debug screen video "chip"

- UDDriveImage.s

  Service routines to read sectors from the host computer. Used for the HD emulation.

13 68k Library examples

Some examples how to use the 68k Library. The samples are located in the 68LibSamples directory.

If you do not want to assemble the examples for your own there are preassembled binaries availble in the bin directory.

You should also have a look to the 68lib sources and check which routines are

available.

Examples:

- BGColorRaster.s (VGA screen must be connected)

  Shows how to set the background colour and how to do some rasters on the background.

- BitmapMode.s (VGA screen must be connected)

  Shows a 128x128 bitmap image

- CustomChars.s (VGA screen must be connected)

  Shows a screen with custom chars and a permanent upload means scrolling char. This example also syncs to the VGA VBL.

- PrintText.s (VGA screen must be connected)

  Simple demo how to print text in different colors to screen.

- Terminal.s (VGA screen must be connected)
  Shows the usage of a partical terminal screen.

14 Troubleshooting

1. **The cartridge is not recognised by the command line tool**

   a) Check if the drivers are installed correctly

   b) Windows: the cartridge should show up under "Devices and Printers" as ultraDev. If not maybe reinstall the drivers and check the USB cable connection.

   c) Mac & Raspberry: Make sure you started ultraDev with sudo ! That's important

   d) If you have an USB floppy emulator, the USB cable is not connected to the floppy emulator and if the software for the floppy emulator is running this can cause this effect. The problem is that the floppy emulator uses the same USB chip unfortunately the authors did not set a unique description for the USB chip means they are scanning every USB FTDI chip if it responces. Great job guys ! So connect the USB cable for the floppy emulator and start the floppy emulator software.

2. **All leds on the cartridge are blinking and the cartridge does not react to anything**

   When all leds are blinking the cartridge waits for a reset from the Atari. There are two ways to solve. 1) Do a reset ;) 2) Press the Config button on the cartridge.

3. **When switched on the cartridge does not react to uploads**

To solve you can press the Config button which resets the FPGA but before you do a photo from the cartridge screen and send me the photo

4. **I changed the font for VGA display but after a reset the font is still changed**

   This is correct. The font is only resetted when the FPGA is resetted means press the Config button. A font reset every time need some time and this would slow down the upload of everything.

   You can reset the font by yourself if you have a look to the 68k library UDFont.s →UDFontReset

5. **I'm sharing a volume and doing a reset with ultraDev but the folder does not appearl. Only if I reset the Atari again**

   This is a known bug and will not be fixed. Just don't do this.

6. **When doing a folder share and when I open a folder the Atari crashes**

   This can happen if in the folder is a very big file. In my case it was a 139mb file. This is no ultraDev bug and seems to be a Tos bug in <=1.62.

7. **When exiting the program with trap #1 the Atari hangs**

   If your start a Prg-File over the command line tool and you leave your prg with:

   ```
   clr.w  (sp)
   trap   #1
   ```

   It will not return to the cartridge screen. Unfortunately not easy to fix that. ultraDev does not use Tos to build up the base page and start the Prg. Without starting via Tos its hard to find out why it hangs. I think I'll not change this.

   During development if you really want to exit do a reset with:

   ```
   lea       $4.w,a0
   move.l    (a0),a0
   jmp       (a0)
   ```

8. **On Mac or Raspberry changes in the directory tree are not recognised**

   This is correct. This feature is currently not supported on Mac and Raspberry. After changes you need to restart the command line tool.

15 Hardware registers

15.1 Reading from and writing to the cartridge

Reading from the cartridge from rom or hardware registers is straight forward just use a move instruction to get the data.

Unfortunately Atari did not add a write signal to the cartridge port. To realise a write to the cartridge you have to use a trick.

So the cartridge are is split into two areas. FAXXXX and FBXXXX. FBXXXX is used for writing.

15.2 Reading

Reading can be done from FAXXXX. Some of the "read" registers act like a switch or a trigger (will be explained a bit below).

15.3 Writing

Writing is a bit confusing I hope I can explain ;)

For writing to the cartridge the area FBXXXX is used. Writing works with a read instruction. The address itself is used for the write value.

XXXX are the bits which are written to the before selected memory destination. So the write value is included in the reading address.

This also means you can not write to a specific address in the destination memory you only can write and the internal address is incremented with each write.

If you want to write to a different memory position you have to set it before.

If you look to the code for writing to the cartridge it does not make sense at the first glance.

Maybe a small example for writing an white "a" to the debug screen:

```
move  UDRegWriteToScreen,d0
move  #('a'+$700)*2,d0
sub.b #$20,d0
lea   $fb0000,a0
move  (a0,d0),d0
```

Looks confusing right? So let's have a look to each line.

```
move  UDRegWriteToScreen,d0
```

Reads from the address $fa4100 the result in d0 does not care. But it does not preform a read inside the FPGA it just switches the write selector to the debug screen memory.

This is what I was talked about earlier. If you read from some memory locations in the FAXXXX area this triggers or switches stuff in the FPGA. In this case it switches the write destination to the debug screen memory.

```
move #('a'+$700)*2,d0
```

Ascii code for an "a" and +$700 is the color for the char. *2 is because the adress lines in the cartridge has no a0 signal so odd access aren't possible.

sub.b #$20,d0

The font starts with $20 (space) so you need to sub $20 from the ascii code.

```
lea  $fb0000,a0
```

Get the write area address to a0.

```
move (a0,d0),d0
```

And read from that address. D0 is in this case the value you want to write. If you read from that address the FPGA does following:

- extracts the XXXX bits from the FBXXXX address
- check what is the write destination
- and writes the XXXX bits to the write destination
- increments the internal write address destination address by 1 (for some write destinations you can also set the increment like for font upload which makes sense in the bitmap mode)

To set the internal address the FPGA writes to is working straight forward. For example setting the write address to the debug screen were to write the char on screen is like:

- Reading from UDRegWriteToScreenPos to select the internal address of the debug screen where to write.
- preform a "write" by reading from the $FBXXXX area to set the address

But ! You do not need to fight with this the 68k library has service routines to handle this.

15.4 Memory map

R/W/S

R = Read

W = Pseudo write via read (explained in 15.3 Writing)

S = Switch memory write destination to...

Address	RWS	Description
FA0000-FA4000	R	"Rom" area
FA4000	R	68k State. Only used to communicate between FPGA and 68k. This is the state what the 68k should for example do during a upload of a program.

FA4002	R	Ready Flag, Only used to communicate between FPGA and 68k. Tells the 68k is for example done with coping the before uploaded block.
FA4004	R	Upload count during upload a program only used for debugging.
FA4006	R	State machine value for the FPGA only used for debugging.
FA4008	R	Bytes received during a upload inside of the block only used for debugging.
FA400A	R	State machine return state value for the FPGA only used for debugging.
UDRegVBLFlag FA400C	R	VGA VBL flag. 0 = currently in vbl 1 not. This is the direct VGA signal which goes to the monitor. Keep in mind this signal will stay at 0 during a vbl for some lines on the VGA screen.
UDRegVGAX FA400E	R	VGA counter x. Current beam x position.
UDRegVGAY FA4010	R	VGA counter Y. Current beam y position.
UDCartDetect1 FA4012	R	Cartridge detect 1 $dead
UDCartDetect2 FA4014	R	Cartridge detect 2 $beef
UDFPGAVersion FA4016	R	FPGA Version
UDDISectorNumber FA4018	R	HD Emulation Sector number to read next
UDDIState FA401A	R	HD Emulation State flags Bit 0 If 1 read sector is busy
UDDIErrorFlags FA401C	R	HD Emulation Error flags (currently unused)
UDDISectorSize FA401e	R	HD Emulation Sector size. The sector size is set by the host application depending on the size.
UDDIReadSectorFlag FA4020	R	HD Emulation Read sector. Initiates sector read.
FA4022	R	Flag if the Cartridge is switched on or off.
UDDIFatSize FA4024	R	HD Emulation Fat size.
UDDIDirSize FA4026	R	HD Emulation Dir size.
UDDICurSector FA4028	R	HD Emulation Current Sector (unused)
UDDIDriveNumber FA402a	R	HD Emulation Drive number of the HD.

FA402c/2e	R	HD Emulation backup of the original hdv_bpb vector.
FA4030/32	R	HD Emulation backup of the original hdv_rw vector.
FA4034/36	R	HD Emulation backup of the original hdv_media vector.
FA4038/3a	R	If the cartridge is switched on here you can find the base address of the cartridge code.
FA403c	R	Cartridge Screen up. Internal use only.
FA403e	R	Cartridge Screen down. Internal use only.
FA4040-FA4052	R	Folder share bpb
UDKeyStroke FA4054	R	Key board stroke from pc. $FF no available. After reading the value is set to $FF again.
UDRegWriteToScreen FA4100	S	Switch write destination to debug screen memory
UDRegWriteToScreenPos FA4102	S	Switch write destination to the internal debug screen address. Used to set where to write on the debug screen.
UDRegWriteToFontMemory FA4104	S	Switch write destination to font memory. Do not forget to set the internal font address before uploading!
UDRegWriteToFontAdr FA4106	S	Switch write destination to the internal font address.
UDRegWriteToFontAdrIncr FA4108	S	Switch write destination to internal auto increment register for font upload. If you upload a font each time to write a byte to the font memory the font address is incremented by 1. In the bitmap mode it's better to set it to 8 to write a horizontal line. If you don't do you need to set the font address each time because the bitmap mode is char based and stuff.
UDRegWriteToReg1 FA4200	S	Switch write destination to video register 1. 0: 0 = 40x32 resolution 1 = 16*16 double pixel mode 1: unused (maybe later multi colour mode) 2-4: background colour
UDRegWriteToSectorLow FA4400	S	HD Emulation write to low value of the sector number.
UDRegWriteToSectorHigh FA4402	S	HD Emulation write to high value of the sector number.
UDRegWriteToSectorCount FA4404	S	HD Emulation write to low value of the sector count (not used currently).
UDRegWriteToSectorBuffer FA4406	S	HD Emulation write to high value of the sector count(not used currently).
UDRegWriteToHdvBackup FA4408	S	HD Emulation write to hdv backup.
FA8000-FAC000		Upload memory for program or data files.