Ultimate Documentation

Gideon Zweijtzer

Dec 25, 2023

CONTENTS

INSTALLING FIRMWARE 3.10I / CORE 1.42

1.1 Instructions

Installing firmware 3.10i works in the same way as previous versions. You simply browse with the file browser in the Ultimate menu to the unzipped firmware file. Move to the appropriate file with the cursor keys and press RETURN to bring up the context menu. Select 'Run Update', and the updater program will be loaded and started.

There are different files in the download package. Which one you need depends on your hardware, and for the 1541 Ultimate-II also on the intended use, as not all features fit together in one firmware version.

It is important to note that the embedded processor of the 1541U2 will change from a Microblaze clone to my own implementation of RiscV. This means that older updaters that were built for Microblaze will no longer run. And this means, that you cannot downgrade from 3.10i using these older ".u2u" files. To make sure you don't load the wrong file, the filename extension that the firmware recognizes changes from ".u2u" to ".u2r". If you want to downgrade, you will first have to make sure your unit runs on Microblaze. This can be done by loading the "revert_to_mb.u2r" firmware, which loads *an unsupported version of 3.10i*, based on Microblaze. Using this unsupported version of the firmware, you can then load any ".u2u" file, including older versions. This is the *ONLY* reason that this unsupported version is provided! See the table below:

1.1.1 1541 Ultimate-II Migration Table

Current Firmware	CPU Type	Upgrade Method	Package / File	Updates to...
Any 2.x	ZPU	'update.bin' on Root of SD Card	1541u2_3.0beta7.zip	Microblaze / 3.0beta7
Any 3.x	Microblaze	Select '.u2u' file from menu	1541u2_3.y.zip	Microblaze / 3.y
Any 3.x	Microblaze	Select '.u2u' file from menu	ultimate_3.10i.zip	RiscV / 3.10i
3.10i and up	RiscV	Select '.u2r' file from menu	update_yyy.u2r	RiscV / yyy
3.10i and up	RiscV	Select '.u2r' file from menu	revert_to_mb.u2r	Microblaze / 3.10i(*)

(*) Version only supplied to allow loading older firmware versions.

1.1.2 What is in the .zip package?

Filename	Purpose	Variant	1541	1571	1581	SID	Sampler	GMod2	ACIA-6551
3.10i/update.u64	Firmware for Ultimate 64								
3.10i/update.u2p	Firmware for Ultimate-II+								
3.10i/update.u2l	Firmware for Ultimate-II+L								
3.10i/update.u2u (*)	Firmware for 1541 Ultimate-II	Audio		–	–			–	–
3.10i/update.u2u (*)	Firmware for 1541 Ultimate-II	Dual Drive		–	–	–	–		
3.10i/update.u2r (*)	Firmware for 1541 Ultimate-II	Audio		–	–			–	–
3.10i/update.u2r (*)	Firmware for 1541 Ultimate-II	Dual Drive		–	–	–	–		
3.10i/revert_to_mb.u2u	Downgrade package to Microblaze	Audio		–	–			–	–
3.10i/revert_to_mb.u2u	Downgrade package to Microblaze	Modem		–	–	–	–	–	
3.10i/revert_to_mb.u2u	Downgrade package to Microblaze	Gmod2		–	–	–	–		–
roms/kernal.bin	Hyperspeed Kernal								
changes.txt	Change Summary / Release notes								
carts	Directory with cartridge files								

(*) Inside of the installer, you can choose which variant of the FPGA will be loaded

1.2 Cartridge Timing

One of the things that often goes wrong is the cartridge timing. With the introduction of the U2+L, the cartridge timing has been revised. This is not always an improvement, as has shown by some experiences from different users. Also, some firmware releases had quite strongly C128-biased settings, which did not work well for many C64s, especially NTSC ones. This is certainly an aspect that still needs improvement, with maybe some automatic calibration.

For best results, for now, choose the following timings. You can find these in the config menu (F2) - under 'C64 and Cartridge settings':

Cartridge	Machine	'PHI2'-timing	'PHI1'-timing
1541U2	C64	100 ns	140 ns
1541U2	C128	200 ns	200 ns
U2+	C64	96 ns	144 ns
U2+	C128	192 ns	192 ns
U2+L	C64	100 ns	140 ns
U2+L	C128	200 ns	200 ns

For kernal replacement, you may need the 80 ns setting for PHI2.

Kernal replacement on a C128 may work for PAL models, with the Shadow RAM disabled, and high timing values

Please help to extend this table to SX-64 and possibly different board revisions.

1.3 Changes in 3.10i, compared to 3.10

1.3.1 Background

If you would take some time to check the commit history of the repository at GitHub, you may find over 500 commits since the last released version, 3.10a. This is a lotttt more than it usually takes to release a next firmware version. The version is still 3.10, so there are not many functional changes. Yet, files have been touched, updated, or just made compatible with the new Ultimate-II+L hardware. It was more difficult than expected to get the Lattice FPGA to work correctly and keep compatibility with the existing hardware platforms.

On top of the new introduction of the Ultimate-II+L, there is still a need to support the "good old" 1541 Ultimate-II (or simply: U2). There have always been issues with the Microblaze processor and the compiler from Xilinx. Every version of the Xilinx compiler had different bugs that caused the Microblaze based firmware to be broken in one way or another. Since the U2+L runs well on the Risc-V, it was decided to upgrade the CPU inside of the U2 and ditch the Microblaze forever. Unfortunately, the CPU that was chosen to run inside of the U2+L did not fit in the same space as the Microblaze clone, so I decided to write my own Risc-V compatible CPU. This is the CPU that will be used from now on in the 1541 Ultimate-II.

1.3.2 General Fixes

- [Important] Fixed ExFAT filesystems with 128kB clusters (Fixes failing D64 mounts from some ExFAT thumb-drives)
- [Facebook Discussion] EasyFlash save function now saves all chip chunks to support incomplete CRT files
- [Issue-271] GEORAM at boot (MarkusC64)
- [Issue-279] Support for 2 MHz operation for any cartridge, including EasyFlash
- [Issue-246] Fixed: Virtual Printer crashes when file cannot be created.
- [Issue-234] Deleting last item in directory makes selection invisible
- [Issue-236] Confirmation modal asking a question only gives 'OK' instead of 'Yes/No'.
- [Issue-R17] Added mapping of ESC key to RUN/STOP.
- [Facebook Request] Added: Leave menu on mount (can be selected per drive)
- [Facebook Bug Report]: Fixed starting Kingsoft Business Basic cartridge
- [Facebook Bug Report]: Allow Maverick (and others?) to write wrong sector headers on MFM tracks.
- [Facebook Bug Report]: Unable to select GeoRAM. Is now a mode of the RAM Expansion Unit.
- [Facebook Bug Report]: Fixed color palette not loaded at boot (U64 only).
- [Facebook Bug Report]: Fixed USB sticks not recognized
- [Messenger Report]: Improvements to palette file read. Rejects faulty files. Fixes hang-up
- [Messenger Report]: Fixes 'Create DNP' when run from Telnet. Turned out to be a stack overflow.
- [Email Discussion]: Adds EEPROM segment to GMOD2 when it is missing in the CRT file

1.3.3 U64 core fixes

- Multicolor graphics side border bug (core version V1.41 => V1.42)

1.3.4 Modem fixes

- ACIA NMI pulse extension to extend over bad line. This allows the 6502 to see it always.
- Performance optimization ACIA / Modem
- Allow spaces after ATDT command.
- Minor fixes in the modem emulation layer from Scott Hutter (xlar54)

1.3.5 Cartridge Support

- [Issue-314] Added support for Blackbox V9.
- Additional C128 cartridge variant that allows banking and offers some RAM (U2/U2+/U2+L only).

1.3.6 Additions

- Firmware target added: Ultimate-II+L; the Lattice version!
- System Information page (F4) now shows elaborate version info.
- GEOS support functions from MarkusC64

INSTALLING FIRMWARE 3.10 / CORE 1.41

2.1 Instructions

Installing firmware 3.10 works in the same way as previous versions. You simply browse with the file browser in the Ultimate menu to the unzipped firmware file. The extension for 1541 Ultimate-II is ".u2u", for the Ultimate-II+ the extension is ".u2p", and for the U64 the extension is ".u64". Move to the appropriate file with the cursor keys and press RETURN to bring up the context menu. Select 'Run Update', and the updater program will be loaded and started. Note for 1541U2 users: *You need to have at least 3.x firmware to do this.*

It is important to know that the organization of the data in the Flash chip changes when installing 3.10. Even when the installer runs, previously stored data may be lost. This is especially the case if you have already used the "/flash" drive (e.g. on the U2+ with the factory installed version 3.9a). So once again, please assume that all user data will be lost. Backup this data if it is important to you. For those who have never seen the "/flash" directory in the file browser; this is a relatively new feature whereby the on board flash chip can be accessed as if it were a disk with files. From 3.10 and onward, non-volatile user data is stored as files on the "/flash" drive.

2.1.1 What is in the .zip package?

Filename	Purpose	1541	1571	1581	SID	Sampler	GMod2	ACIA-6551
update_1.41.u64	Firmware for Ultimate 64							
update_3.10.u2p	Firmware for Ultimate-II+							
update_3.10_au.u2u	Firmware for 1541 Ultimate-II		–	–			–	–
update_3.10_dd.u2u	Firmware for 1541 Ultimate-II		–	–	–	–	–	
update_3.10_gm.u2u	Firmware for 1541 Ultimate-II		–	–	–	–		–
uci_kernal_3.10.bin	Updated Hyperspeed Kernal							
changes.txt	Change Summary since 3.9/1.37							

Below, as a reference, is a list of the most important changes since the previous official release 3.9. Documentation for the most prominent new features or changes in user interface can be found here:

- Multi Mode Disk Drive

- Updated Cartridge System

- Programmable U64 Palette

2.2 Changes since official release (3.9 / 1.37)

2.2.1 Added features

- Hardware emulation of 1571 and 1581 drives (U2+ / U64 only)

- MFM support on 1571, with enhanced G71 format to store MFM tracks

- Added audio samples for insert / remove floppy disk

- Added second drive sound to speaker output

- Possiblity to use your own drive sound samples

- Custom U64 palettes

- SystemInfo Page, showing drive, cartridge and storage status (under F4 key)

- Flash Disk, to store drive roms, sounds, cartridges and such

- Cartridges are now always CRT files. Selected by filename in the config

- Cartridge compatibility check and reporting

- GMOD2 support, including EEPROM (U2+ / U64 only)

- Zaxxon Cartridge support

- Implemented writing CVT files back to a disk image (enables copying CVT files across disks.)

2.2.2 Various UI Improvements

- Select & Delete, using Shift-DEL

- Recursive delete from context menu

- Long filename truncation (thanks to 'naali' / Antti Svenn)

- Scrollable string edit box

- Adds save function for all cartridges, including EasyFlash and GMOD2

- Full Clear function of configuration in Flash

- Added 'F3' Help screen to config menu

2.2.3 Technical Fixes

- Fixed RGB mode (U64 only)

- Fixed block read command

- Fixed access of files with special chars in CBM disk images, accessed from IEC

- Fixed UCI issues with SidPlay and others

- Fixed colon issue in Ultimate Kernal ROM

- Fixed crash on invalid file chain CBM filesystems

- Fixed check order DIR / VOLUME; fixes some issues with exFAT directories

- Added M and V commands in modem emulation layer, V now supporting numeric responses. Thanks Scott Hutter

- Fixed the root cause of the VOLUME bit set in exFAT directories

- Fixed missing 'probe' for filesystems on multi-partition disks

- Fixed loading from wrong device ID when issuing the 'run disk' command

- Fixed many little bugs in the generation of CVT files (copying GEOS files from a disk image)

- Adopted changes to U64 Kernal to set default loading ID, as submitted by Leif Bloomquist

- Adopted USB modifier key handling, as submitted by Peter de Schrijver

- Corrected bug in extended partition table reading

- Relaxed C128 cartridge timing (for C128 carts)

- Extended UCI I/O bytes to support Hyperspeed Kernal properly

- Check for file copy onto itself

- Increased robustness against loading faulty disk images

- Fixed VIA latch mode in 1541/1571.

- [Issue 227] Fixes 'get sensible name' function for Ulticopy

- Fix for .d71 sector allocation - format should give 1328 blocks free, not 1347.

- Fixes possible crash when decoding a GCR track to binary

- [Issue 160]: SpeedDOS / DolphinDOS loading errors

QUICK GUIDE TO THE ULTIMATE-II+

Thank you for purchasing the "Ultimate-II+" cartridge: The versatile storage solution for your Commodore 64 computer. Although the installation and use of the Ultimate-II+ are designed to be intuitive, this quick guide shows you the basic functions.

3.1 Installation

1) Take a USB stick, and use a PC or Mac to transfer your favorite floppy disk images (.D64 or .G64 files), tape archives (.T64 files), tape images (.TAP files), Amiga MOD files, SID music (.SID files), or separate programs (.PRG files) to this USB stick.

2) Insert the USB stick into one of the available USB slots on the Ultimate-II+.

3) Take the Ultimate-II+, and carefully slide it into the cartridge slot of your C=64. Connect the supplied serial cable between the serial port connector to the serial port of the computer. In case you want to use a configuration with a real drive, connect the computer to the drive first and then use the feed through of the drive to attach the Ultimate-II+ as the last device in the chain.

3.2 Ports and buttons
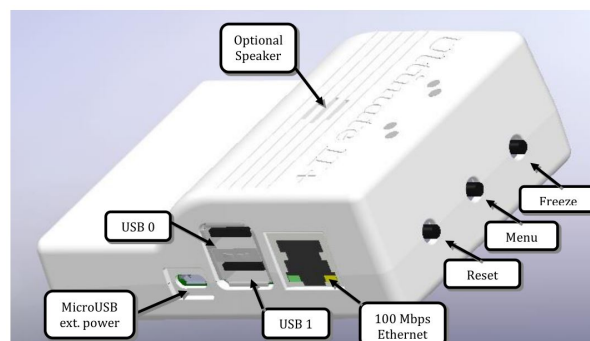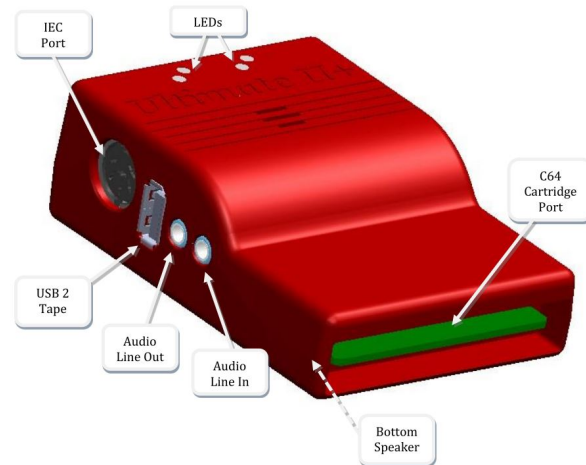
3.3 Concept of operation

The main feature of the Ultimate-II+ is a cycle accurate implementation of the Commodore 1541 Floppy Disk Drive. This part of the Ultimate-II+ cartridge thus acts exactly like a real drive. It is not any faster or any slower than a real floppy disk drive.

Just like a real 1541 disk drive, the Ultimate-II+ needs "floppy disks". On the Ultimate-II+, these floppy disks are *virtual*. These virtual disks exist in the form of a .D64 or .G64 file on a storage device, such as a USB stick. When the user selects such a file, it is linked to the emulated drive. This process is called *'mounting'*. After mounting (which will take a few seconds to complete), the 1541 can access the image file as if it was a real floppy. All standard commands that you are familiar with while using a real CBM1541 then apply.

When necessary, the disk drive can be switched into 1571 and 1581 mode as well. This will happen automatically when the associated file types are selected: Mount a .D71 / .G71 file and the user will be prompted to change the mode of operation to 1571. Similarly for .D81 files, the drive will switch to 1581 mode.

3.4 The Menu

The Ultimate-II+ provides a user friendly menu driven interface, which can be accessed by pressing the middle button on the back of the cartridge. Pressing this button will interrupt the currently running program on the C=64 and show the menu. Leaving the menu will resume the C=64 gracefully. The main two functions of the menu are: 1) file selection, 2) configuration of the 1541 Ultimate cartridge.

The menu starts with showing the available network interfaces, and the storage devices that are attached to the cartridge. By default, it will only show the built-in Ethernet port. When USB-sticks are inserted, one or more items will appear in this list. This screen is the 'root' of the file system. Use the cursor keys to navigate through the file system and select your file (disk image) to mount. The highlighted line shows the current selection.

Use the keyboard as follows:

Key	Function
CRSR up/down	Move the cursor (highlighted line) up/down
CRSR left	Go one level up (leave directory or disk)
CRSR right	Go one level down (enter directory or disk)
Return	Bring up a context menu with actions to be performed on the selected item.
F1	Page up
F7	Page down
F5	Bring up a menu with actions to be performed.
F2	Enter the setup menu
F3	Shows the HELP-text
F4	Shows the current system configuration
Space	Select / deselect entries in the file browser for copying.
C=C / C=V	Can be used to copy files between locations, just like in Windows. Be aware that file copy is a lot slower than on a modern PC.
RUN/STOP	Leave the menu.
Quick seek:	Use the keyboard to type the name to search for. You can use '?' as a wildcard.

3.5 More about mounting disks

Remember that 'mounting' a disk means creating a link between the 1541 part of your ultimate cartridge, and a file on a storage device. This means, that when the link has been established, "writes" that the 1541 performs will be written back into the disk image file as well. In this way, the 'real' behavior of a floppy is emulated.

This behavior will not always be what you want. There are two options to avoid this: 'mount' the floppy as read-only, or 'mount' the floppy in 'unlinked' mode. When write protected, the drive itself cannot write to the floppy. In unlinked mode, the writes to the floppy are performed, but do not reflect in the .D64 or .G64 file. When the storage device is removed, the 1541 drive automatically switches to unlinked mode. Be aware that writes to the disk will then be lost when you turn off the machine.

3.6 LEDs

The Ultimate-II+ has four LEDs that are visible on the outside of the plastic case:

  a. Green: Drive power

  b. Red: Drive status

  c. Yellow: Cartridge active

  d. Green: Storage activity

The drive power LED has four states:

  • Off: The drive is not powered

  • Dim: Powered, but no disk mounted

  • Half bright: Powered and disk mounted, motor off

  • Bright: Powered, disk mounted and motor running.

3.7 Cartridge Emulation

Because the Ultimate-II+ will occupy your cartridge slot, a list of popular expansions available for the C=64 can be configured. Think of extensions like Action Replay, Retro Replay, The Final Cartridge III, Super Snapshot V5, KCS Power Cartridge, the Epyx fastloader and many more. In the configuration menu, even the CBM1750/1764 RAM Expansion unit can be enabled, with expansion RAM up to 16 MB!

Information on how cartridge emulation is set up can be found in this section *about cartridges*.

Many of these cartridges have a reset button and a freezer button. This is the function of the other two buttons on the Ultimate-II+. By default, the left button is the freezer button, and the button on the right is the reset-button. Through the configuration menu, the buttons can be swapped if desired.

3.8 Configuration

As mentioned above, you can enter the configuration menu by pressing 'F2' while you are in the menu structure. This will bring up a screen with the following items:

- Clock Settings
- Audio Output Settings
- Software IEC settings
- C64 and cartridge settings
- User Interface Settings
- Drive A Settings
- Drive B Settings
- Network Settings

Use the UP/DOWN cursor keys to navigate and RIGHT to enter the desired configuration screen. Once inside a settings screen, the behavior of the keyboard is slightly different:

Key	Function
CRSR up/down	Move the cursor (highlighted line) up/down
• / -	Increase or decrease a setting, cycling through the available options.
DEL	Go one level up
Return / Space	For string fields: bring up a text entry box For enumerated fields: bring up a context menu with the available options
Run-stop[2]	*Leaves* the configuration menu and saves the new values.

Note that some values will only take effect after reset or a power cycle of your C=64. However, the main menu provides a way to restart the 1541 and the C=64 with the new settings.

---

[2] On VT-100, use backspace instead of RUN STOP

3.9 Ethernet

Some of you may be familiar with the RR-net solution that brings Ethernet to the C-64. Currently, the built-in Ethernet port does *not* provide RR-net compatibility. However:

The Ethernet port is used natively by the firmware. There is some primitive support for file-transfer using FTP, and it is possible to connect to the Ultimate-II+ using a VT-100 terminal program on the Telnet port (port 23). This gives the possibility to control the machine remotely, and swap disks without actually interrupting the program running on the C-64.

3.10 Modem support

Starting from version 3.7, a lightweight modem emulation layer is provided in the firmware of the Ultimate. This modem layer is accessible through an emulated MOS 6551 ACIA chip. This chip was found in the SwiftLink cartridge, as well as some other ACIA based RS-232 cartridges of the time.

The modem emulation layer bridges the ACIA chip and the LAN port. The modem connects to a server (e.g. one that is running a bulletin board system) through the internet. In order to use this, the modem needs to be enabled in the configuration menu of the Ultimate application, as well as in the terminal program, e.g. CCGMS. The modem will respond to commands such as "ATDT", which is followed by the domain name, followed by a colon and the port number. For example:

ATDTAFTERLIFE.DYNU.COM:6400

Incoming connections are also supported. This enables you to run a simple server on your C64, or maybe even a BBS!

3.11 USB support

The Ultimate-II+ supports most USB sticks and Flash card readers out of the box. It also supports USB 2.0 Hubs. It is recommended to use powered USB hubs only (with an external power supply). USB 1.1 Hubs are *not* supported.

There are three USB 2.0 ports available on the cartridge; two on the right and one on the left side. Please note that the port on the left is **not** a USB 3.0 port. Although you may use this port as a 2.0 port, it is not USB 3.0 compliant. The extra signals on a USB 3.0 connector are used to communicate with the tape port (see below). **Please do not attempt to attach a USB 3.0 device to the blue USB connector on the Ultimate-II+.** You may safely use a USB 3.0 device on the *right* side of the cartridge.

3.12 File Systems

Currently, the Ultimate-II+ supports the FAT16/FAT32 and exFAT file system on any storage device, and the ISO9660/Joliet on CD/DVD ROM drives, or ISO files. It is able to read D64 files, as well as D71, D81 files (no partitions) and DNP files, including sub directories, T64 files.

3.13 DMA loads

The Ultimate-II+ is capable of loading files directly through the cartridge port into the memory of your C=64. This is called DMA load. The menu supports loading files of the .PRG type only. It doesn't matter if the PRG is located on the FAT/ISO file system, or inside a disk image (.D64), or inside a tape archive (.T64). Be aware that a lot of programs inside a .D64 file require that the rest of the disk is mounted in the drive. For those programs, use the 'Mount & Run' command.

Important! Because the DMA load requires a bit of software on the C64, a special 'loader cartridge' is enabled to perform the DMA. This effectively *disables* the configured cartridge, so you will lose the speed loader capabilities. If you do not want this, you can go to the basic prompt with your cartridge active, and then select "DMA" on the .PRG file in the menu. This will not invoke the cartridge, but simply attempts to load the program into memory. This will not work for all PRGs, however.

3.14 Tape Support

The Ultimate-II+ is capable of emulating a cassette tape deck (CBM1530/1531). To use this function, a special adapter set connects the Ultimate-II+ to the tape port of your C=64 computer. This tape adaptor set can be purchased separately.

In order to play a tape, browse in the menu to a .TAP file, and press enter and select "Play Tape" from the popup menu. This will initialize the tape streamer from the start of the tape. Use functions in the main menu (F5) to pause/resume playback. It is also possible to write the .TAP file to a real tape, using a CBM1530/1531 deck.

The Ultimate-II+ can also capture tape signals into a .TAP file. The 'F5' menu will show you the available options.

3.15 Audio

The green audio connector on the left side of your device provides a stereo line-out signal. In the configuration menu, it can be selected what is routed to the output channels. Available options include:

- Stereo SID;
- Ultimate Audio module (for playing samples);
- Tape read/write pins (for listening to the tape beeps);
- Drive sounds.

Technical note: The emulated stereo SID is taking the CPU writes from the cartridge slot. Unfortunately, there is no way to tell whether the access is made to the I/O region ($D400-$D7FF), or to the RAM below. The necessary signal to differentiate between the two accesses is simply not available on the cartridge port. For this reason, when software uses the RAM in this area, it is possible that you will hear unintended clicks and pops, or even notes.

The blue line-in connector may be used to mix external audio with the output of the Ultimate-II+ cartridge. The separate left and right inputs can be found in the Audio Mixer configuration menu.

3.15.1 Ultimate Audio module

The Ultimate Audio module provides 8 simultaneous sampling voices. This module is utilized as part of the Ultimate-II+ firmware for playing Amiga MOD files. This option is available in the context menu in the file browser.

If you like to do some programming yourself using this sampler, you can enable this module in configuration menu. It then appears in the I/O region. The programming interface is fully documented. Documentation can be downloaded from the official website:

https://github.com/GideonZ/1541ultimate/blob/master/doc/ultimate_audio_v0.2.pdf

3.16 Alternate ROMs

The Ultimate-II+ allows you to use other ROMs for both the emulated 1541 drive, as well as the built-in Kernal ROM of your machine. These ROMs are resident (remembered) by the Ultimate-II+ once they are loaded from the file system.

In order to use an alternate ROM, browse in the file system to the binary ROM file that you like to use. The file should have the '.bin' or '.rom' extension. When you press enter, and the file is of the correct size, the option will appear: "Use as..". Kernal ROMs should be exactly 8K, and drive ROMs should be exactly 16K or 32K.

NOTE: When you use an invalid file as Kernal replacement, the C64 will no longer boot. However, even when the C64 gives a black screen, you can still enter the configuration menu in order to disable the Kernal replacement option.

3.17 Software IEC

The Software-IEC module is a serial bus service that can be enabled in the configuration menu. This module provides two additional devices on the Commodore serial bus; the IEC bus:

- Virtual drive that gives direct access to the Ultimate-II+ file system;
- A virtual printer

3.17.1 Printer

The virtual printer is a valuable contribution created by René Garcia. It takes printer commands from the Commodore 64, and creates a black and white image of the printed graphics and text. This image is then saved to the USB flash drive. The full documentation of the printer emulation and all of its capabilities and options is available here:

https://github.com/GideonZ/1541ultimate/blob/master/doc/ultimate_printer.pdf

3.17.2 Virtual drive

The virtual drive can only be used to access files of the file system, through the OPEN/CLOSE commands on the IEC bus. By default, the path of the IEC drive is '/Usb0', which is the top most USB connector on the right of the unit. This default path can be changed in the configuration menu. When the USB drive contains a program 'TEST.PRG', it can be loaded with the basic command LOAD"TEST.PRG",10. Similarly, you can save your programs with the SAVE command. When loading the directory (LOAD "$",10), the path will be shown as disk name.

The command channel 15, can currently only be used to change the current directory. Just like on modern systems, ".." is the parent directory and "/" is the root directory:

OPEN 15,10,15,"CD:/USB1/MYPROGRAMS":CLOSE 15

At this point, the virtual drive is not JiffyDOS compliant.

3.18 Ultimate Command Interface

Since some time, it has become possible to control the Ultimate-II+ programmatically through the I/O port of the C64, thus from a program that runs on the machine. This is useful for many things; for example it can be used to access the file system much faster than through the serial bus. But it can also be used to load files from the file system into REU memory for example. The set of commands grows over time and will provide more and more powerful features.

Documentation of the interface itself is available here:

https://github.com/GideonZ/1541ultimate/blob/master/doc/Command%20Interface%20V1.0.pdf

Accessing the file system is done through the "DOS" target, which is documented here:

https://github.com/GideonZ/1541ultimate/blob/master/doc/ultimate_dos-1.2.pdf

3.19 Real Time Clock

For getting correct time stamps on the files that are created on the USB pen drives, the Ultimate-II+ offers a real time clock (RTC) function. This RTC can be set through the configuration menu.

The RTC is powered by a CR2032 battery, which is located inside of the unit. Calculations show that the lifetime of this battery is several years.

3.20 Social Media

For quick answers to many questions regarding your device, you may be interested to join the the Facebook group "1541 Ultimate".

3.21 Firmware Updates

In order to update the firmware, you will need a file with the ".U2P" extension, or ".U2L" for the Ultimate-II+L. Such a file can be found in the '.zip' archives found in the download section of the http://ultimate64.com website, after logging on to the site.

Procedure: Use the Ultimate-II+ file browser to find the .U2P / .U2L file. Press ENTER, and the option "Run Update" will appear. Select this option and follow the instructions, if any. After running an update, the unit will reset itself completely after about a minute, without any notification. This is normal behavior.

When, for some reason, the update has failed and corrupted the Flash chip; you can start the "Recovery Mode", by holding the middle button while powering on the unit. In the recovery mode, not all features are available, but the "Run Update" feature should work. (Not available on U2+L!)

3.22 Disclaimer

The 'firmware' on your Ultimate-II+ board consists of a rather large number of functional parts that all work together. Although an insane number of hours have been put into testing and improving the firmware and software, I am very certain that it still has bugs. Some testing work still needs to be done. The device will be improved further as time passes. Check the download section at the website for the newest firmware version. We believe in the end this will sure be the 'ultimate' storage solution for your 8-bit Commodore computer.

3.23 Known issues

- The processing speed of the Ultimate-II+ is currently (V3.10) still quite a bit lower than the Ultimate-II. This is because the Ultimate-II+ platform does not have instruction/data caches still on its embedded processor. The lower speed may cause some situations in which the unit appears to hang, while it is just busy. For example, when using the printer emulation, the conversion from a bitmap image to a PNG file can take quite some time. This may be solved in a future software upgrade.

Other issues are reported and maintained on GitHub:

https://github.com/GideonZ/1541ultimate/issues

ULTIMATE-64 DOCUMENTATION

Welcome to the landing page of the Ultimate-64 documentation!

If you are unfamiliar to the Ultimate range of products, it is recommended to read the *Quick guide to the Ultimate-II+* first, which was written for the Ultimate-II+ cartidge. The application that manages the Ultimate-64 is in essence the same as the Ultimate-II+. Features and functions of this cartidge are also present in the Ultimate-64.

4.1 Getting Started

What do I need to get started?

4.1.1 Prerequisites

- Ultimate 64 motherboard
- Power adapter 12V DC (included)
- A case to put your Ultimate 64 in
- A C64 keyboard
- HDMI cable or standard C64 video cable
- USB Pen drive
- Kernal ROM (extract from your Commodore 64 or download online)
- Basic ROM (extract from your Commodore 64 or download online)
- Char ROM (extract from your Commodore 64 or download online)

Optional

- One or two SID chips
- Keyboard risers (for the C64C model case)

4.1.2 Installing

Mounting the Ultimate 64 in a Case

If you have a spare Commodore 64C case, or bought a new case (which is the C64C model as well), you will need keyboard risers. Some of these have loose metal keyboard risers already in them, others have them mounted with pop rivets to a metal shield. If you have the latter, it is recommended to drill the pop rivets to get the risers loose, otherwise they will interfere with the keyboard connector on the Ultimate 64.

A few C64C have the keyboard clipped onto the top of the case. For those you don't need keyboard risers.

If you buy a new case online, be aware that you will need a keyboard as well.

If you have an older case (the breadbin model), the keyboard is mounted in the top of the case, and therefore you don't need keyboard risers.

1. Mount the risers to your keyboard. The left can be farthest to the edge, but the right riser you have to mount a bit to the left of the right edge, so it won't interfere with the keyboard connector.

2. There are some places online where you can find risers that are supposed to fit perfectly with the Ultimate 64. Here is one model that should work: https://www.thingiverse.com/thing:3051450 You can 3D-print them yourself or ask someone to do it for you. These can be screwed directly onto the bottom of the case as well.

3. Set the keyboard aside for now.

4. Now gently slide the Ultimate 64 board in your case, and fit it to the right side of the case first. The joystick connectors should slide in easy.

5. Push it gently against the back of the case, and let the connectors on the back of the board fit through the holes.

6. Adjust the board so you can see the mounting holes on the bottom of the case through the Ultimate 64 board mounting holes.

7. Fasten the board with suitable screws. Don't tighten too much.

8. Put the keyboard in the case, and connect it to the connector to the right. It will only fit one way, as there are a key pin on the connector.

9. Attach the top of the case.

Accessories

If you have a 3D printer, or you know someone who has, you can find many nice accessories at thingiverse. Here are a few of them:

- Ultimate64 Keyboard Mounting Brackets
- Ultimate64 & C64 Reloaded power connector cover
- Ultimate64 Multi-Button (Power Button) cap
- Ultimate64 board support clips

Commercially available kits

If you do not have a 3D printer yourself, you may also want to take a look at these kits that are available from Corei64 by Thomas Christoph, or the Pixelwizard Retro Shop from Thomas Koch. Both offer professional kits that make mounting of the U64 easy. Both shops have put special pages together, just for the U64, where the items are listed:

- All Ultimate64 accessories from Corei64

- All Ultimate64 accessories from Pixelwizard

- Installation kit for 64C Corei64

- Installation kit for 64C Pixelwizard

- Installation kit for breadbin Corei64

- New C64C cases in different colors Pixelwizard

Other installation accessories, stickers and batches are available at https://www.corei64.com/shop/ or at https://shop.pixelwizard.eu/en

Connecting your Ultimate 64

Connect your Ultimate 64 to the wall and your TV/monitor. If you start with HDMI, your TV/monitor have to be DVI capable over HDMI. You can change this to HDMI later.

Flashing ROM

Before you can use your Ultimate 64, you have to flash your Kernal and Basic ROM to the Ultimate 64 (keep in mind that some alternative kernal ROM's removed tape routines). Power on and wait for picture, where it explains how to do.

Basically you stick your USB drive in the back of the Ultimate 64 and push shortly on the power button to bring up the Menu.

The ROMs has to have the suffix `.bin` or `.rom`, as in mykernal.bin, mybasic.bin, mychar.bin.

1. Browse to your Kernal ROM image and press Enter.

2. Choose `Flash as Orig. Kernal ROM`.

3. Browse to your Basic ROM image and press Enter.

4. Choose `Flash as Orig. Basic ROM`.

5. Browse to your Character ROM image and press Enter.

6. Choose `Flash as Orig. Character ROM`.

7. Push `F5` and choose `Reboot C64`.

Real SID on Board revisions 1.1, 1.2 and 1.2a

If you're installing real SID chips, you have to set the jumpers accordingly. If you don't have or don't want to install SID chips you can use the built in 'UltiSID'.
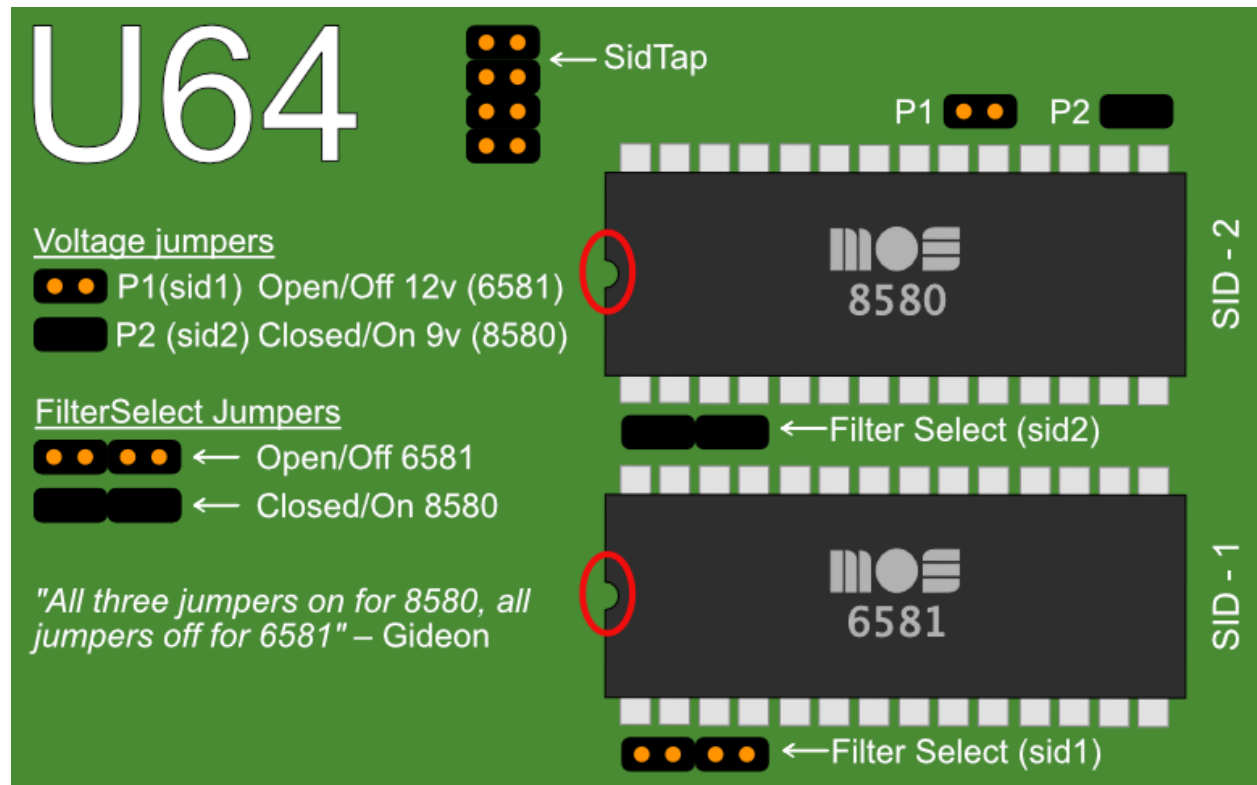

```
Voltage Jumpers:
P1: SID 1 Voltage
P2: SID 2 Voltage
On/Closed: 9V (8580)
Off/Open: 12V (6581)

Filter Select Jumpers:
On/Closed: 8580
Off/Open: 6581
```

In Firmware revisions before V1.22, it is required to set your SID types in the Ultimate-II+ menu:

```
Short press on power button
F2
U64 specific settings
Set your types at
SID in socket 1
and
SID in socket 2
```

Setting the SID type in the configuration enables the bus access to the chip, and also allows the auto-configurator to

choose the right chip when playing a SID tune with the provided player from Wilfred Bos.

Real SID on Board revisions 1.3 (U64 Elite) and 1.4

The newer board revisions are jumperless. This means that both the voltage as well as the filter caps are selected automatically, based on the inserted SID type. For this, firmware V1.22 or later is *required. Do not downgrade to earlier versions on jumperless boards!*

In order to make sure that 8580 chips are never given a higher voltage than they can handle, the socket gets disabled automatically whenever an 6581 chip (which needs 12V) is detected for the first time. When the popup appears "SID changed", enter the configuration menu, and enable the socket in which the 6581 resides:

`Short press on power button`
`F2`
`SID Sockets Configuration`
Enable the sockets with options:
`SID in socket 1`
and
`SID in socket 2`

Once the chip in the socket is recognized as 6581 AND the socket is enabled, the 12V is applied.

Pull Down Resistor

Note that in the original schematics of the C64, a so called "pull-down" resistor of 1 kΩ is placed from the audio output pin to ground. This resistor is present on the 6581-based boards, but absent on the 8580-based boards. On the jumperless U64 boards this resistor is also enabled and disabled automatically. However, you may override this in the same configuration menu. The influence is limited.

4.2 Multi Button

The Ultimate-64 has one button on the side of the board, which is placed on the original position of the power button.

This button is called the MultiButton, because it handles several functions.

Function	Description
Power on	When the machine is off, press the button briefly to switch it on
Power off	Press the button for 4 seconds to switch off the machine.
Reset	Press the button for about 1 second to reset the machine.
Menu	Press the button briefly to enter the File Selection (Ultimate) menu.
Freeze	Hold the button and briefly tap on the RESTORE key to get into the freeze function of the selected utility cartridge.
Config Reset	With the machine switched off, press the RESTORE key and keep it pressed. Then turn the U64 on by briefly pressing the button.

4.3 User Interface

The Ultimate-64 provides a user friendly menu driven interface, which can be accessed by pressing the multi button (power) briefly.

There are two modes in which this user interface can operate:

1. Freeze mode
2. Overlay mode

In Freeze mode, pressing the power button briefly will interrupt the currently running program on the C=64 and show the menu. Leaving the menu will resume the C=64 gracefully. This mode of operation works on both PAL and HDMI outputs.

In Overlay mode, pressing the power button briefly will show the menu in an overlayed area on the screen. The C=64 will simply continue running, but the keyboard is switched over to the Ultimate application, as long as this overlay screen is visible. Pressing run-stop in the file browser, or pressing the power button again briefly will make the overlay disappear.

The user interface has three main functions:

1. File browser
2. Configuration of the Ultimate 64 (F2)
3. Tool menu (F5)

4.3.1 File browser
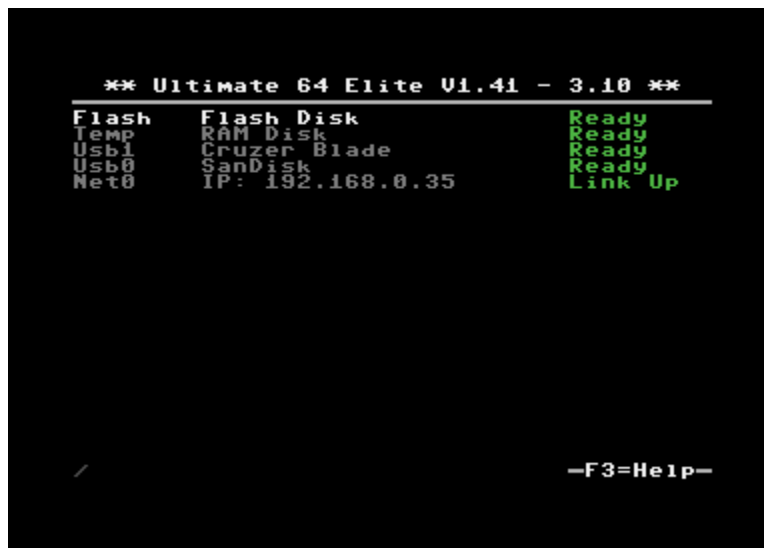
The menu starts with showing the available network interfaces, and the storage devices that are attached to the cartridge. By default, it will only show the built-in Flash drive, a RAM disk for temporary file storage and the built-in Ethernet port. When USB-sticks are inserted, one or more items will appear in this list. This screen is the 'root' of the file system.

Use the cursor keys to navigate through the file system and select your file (disk image) to mount. The highlighted line shows the current selection.
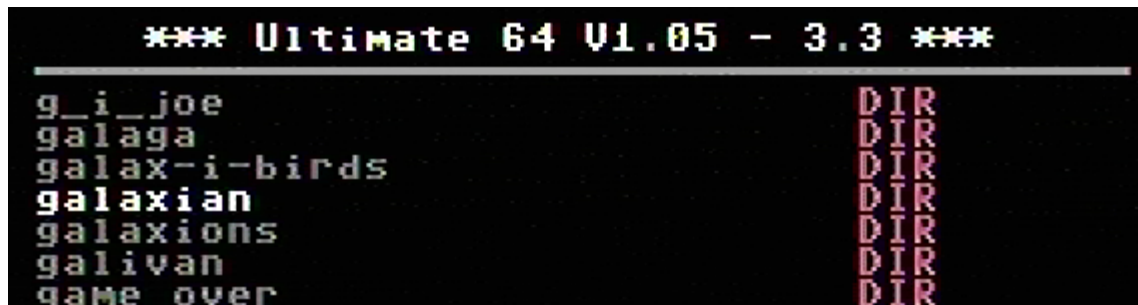
4.3.2 F2 - configuration menu

Use the UP/DOWN cursor keys to navigate and RIGHT to enter the desired configuration screen. Once inside a settings screen, the behavior of the keyboard is slightly different:

Key	Function
CRSR up/down	Move the cursor (highlighted line) up/down
• / -	Increase or decrease a setting, cycling through the available options.
DEL / CRSR left	Go one level up
Return / Space	For string fields: bring up a text entry box For enumerated fields: bring up a context menu with the available options
Run-stop	*Leave* the configuration menu.

Note that some values will only take effect after reset or a power cycle of your C=64. However, the main menu provides a way to restart the 1541 and the C=64 with the new settings.

4.3.3 F5 - Tool / Command menu

The purpose of the tool menu is to command the Ultimate application to do something that is not related to the selected entry in the file browser menu. For instance, creating a new disk image or directory is such an 'action' that can be performed using the tool menu. Some options might appear / disappear, or be greyed out depending on the state of the application, or based on whether the current file browser directory is writable. For instance, `Create Directory` is available only when the file browser is in a location were directories can be created.

More information about each of the available options can be found in the description of the Ultimate Application itself, with all of its features.

4.3.4 The Joystick switch on Ultimate 64 Elite boards

The Ultimate 64 Elite bords have the additional feature to swap the joystick ports. This feature is useful when playing different games, each one expecting the joystick on a different port. If you have only one joystick plugged in, you can use it to change the port at a time. To change the port first press the power button. Now the file selection screen will pop up. now press the Commodore key and the "J" key on the keyboard. The Ultimate 64 Elite now will automatically close the menu and jump back to your C64 screen. Repeat the steps above to again switch the port.

4.4 Configuration menu

Within this menu the U64 settings can be customized.

4.4.1 Audio Mixer settings

Volume and panning for various sources can be configured in this menu.

Note that these settings affect the HDMI and AV outputs only. They have no effect on the levels of the SID-TAP header which takes signals directly from the SID sockets.

Also note that if you have enabled the **SID Player Autoconfig** option in **U64 Specific Settings** menu, using the built-in SID player will override the SID settings you have configured in the audio mixer.

The volume level of the optional internal speaker can be adjusted in **U64 Specific Settings** menu with option **Speaker Volume (SpkDat)**.

Volume options: OFF, -42 dB - +6 dB (0 dB is normal level, OFF is disabled)

Panning options: Left 5-1, Center, Right 1-5

Menu name	Explanation	Default value
Vol UltiSid 1	Set volume for UltiSID 1	0 dB
Vol UltiSid 2	Set volume for UltiSID 2	0 dB
Vol Socket 1	Set volume for SID in socket 1	0 dB
Vol Socket 2	Set volume for SID in socket 2	0 dB
Vol Sampler L	Set volume for left channel of Ultimate Audio module	0 dB
Vol Sampler R	Set volume for right channel of Ultimate Audio module	0 dB
Vol Drive 1	Set volume for drive 1 (A) sounds	-4 dB
Vol Drive 2	Set volume for drive 2 (B) sounds	-4 dB
Vol Tape Read	Set volume for tape read sounds	-36 dB
Vol Tape Write	Set volume for tape write sounds	-36 dB
Pan UltiSID 1	Set panning for UltiSID 1	Center
Pan UltiSID 2	Set panning for UltiSID 2	Center
Pan Socket 1	Set panning for SID in socket 1	Left 3
Pan Socket 2	Set panning for SID in socket 2	Right 3
Pan Sampler L	Set panning for left channel of Ultimate Audio module	Left 3
Pan Sampler R	Set panning for right channel of Ultimate Audio module	Right 3
Pan Drive 1	Set panning for drive 1 (A) sounds	Left 2
Pan Drive 2	Set panning for drive 2 (B) sounds	Right 2
Pan Tape Read	Set panning for tape read sounds	Center
Pan Tape Write	Set panning for tape write sounds	Center

For example, if you want to make sure that you only hear the SID in socket 1 and no other real or emulated SIDs, you might adjust the mixer settings like this.
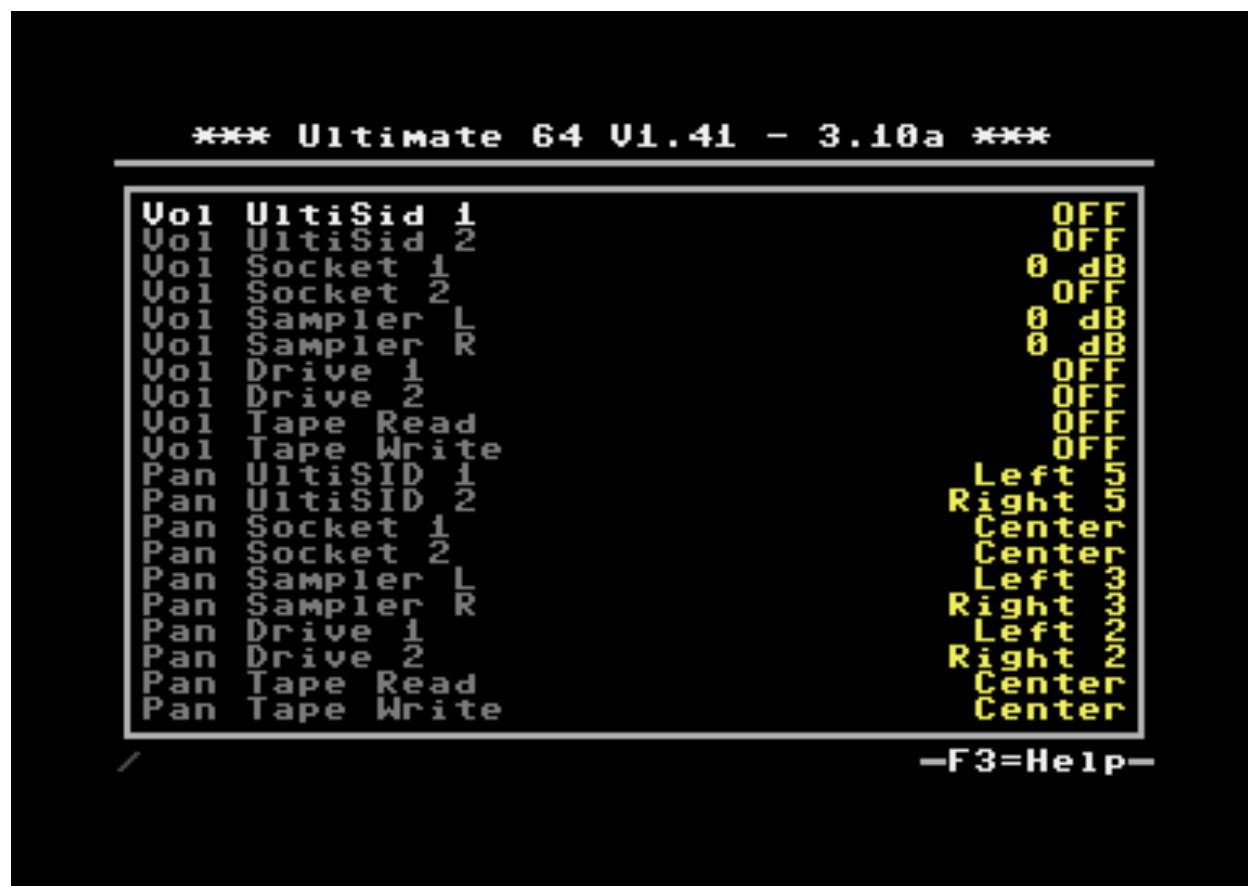
In this example the drive and tape sounds are also disabled, panning for both SID sockets is set to center and the UltiSIDs are panned hard left and right. Of course those panning settings don't matter here because you can only hear the sources where volume is not set to OFF.

4.4.2 Turbo mode settings

Turbo control
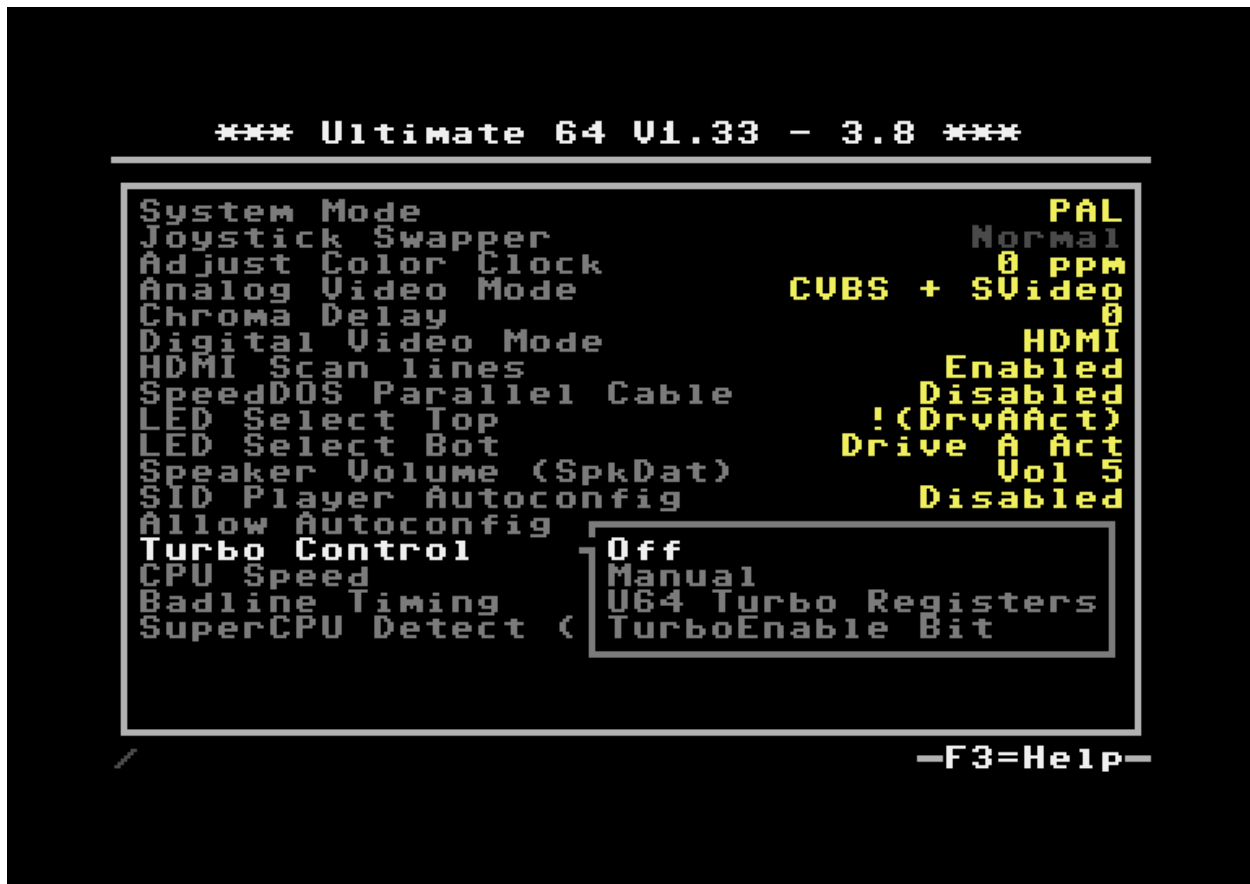
Option	Description
Off	Turbo control is off
Manual	Turbo control uses settings from U64. Settings **CANNOT** be altered with registers.
U64 Turbo Registers	Turbo control uses settings from U64. Settings **CAN** be altered with registers. *See Turbo Control registers.*
TurboEnable Bit	Turbo control uses settings from U64. Settings **CAN** be altered with registers. *See Turbo Control registers.*

CPU Speed

Set CPU Speed.

Default C64/U64 speed: **1MHz (1x)**
Maximum speed: **48MHz (48x)**

Badline Timing
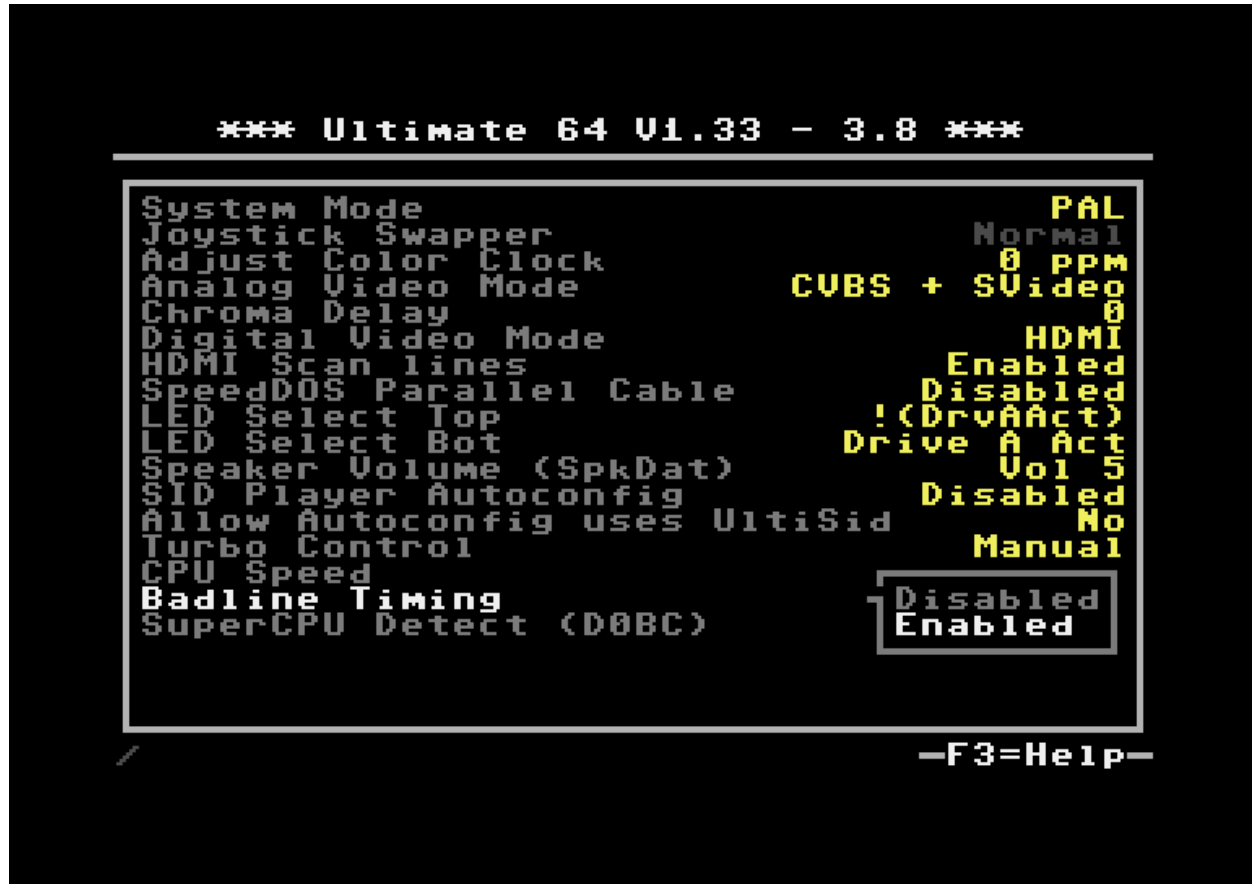
Enabled/disable badline timing.

Turn off badlines to get more CPU cycles.

This can be controlled with registers programmable. See Turbo Control registers.

SuperCPU Detect ($D0BC)

SuperCPU Mode Detect Register

More information: https://www.c64-wiki.com/wiki/SuperCPU

4.4.3 Turbo Control registers

Address	Description

53297/$D031	**U64 Turbo Control**
	This register is only available when the Turbo Mode selector is set to 'U64 Turbo Registers' or 'Turbo Enable Bit'. Otherwise it simply reads $FF.
	  • bit 0-3: CPU Speed (index)
	  • bit 7: Badline timing, 0 = Enabled, 1 = Disabled
	CPU Speed (index)
	  • 0 = 1 MHz
	  • 1 = 2 MHz
	  • 2 = 3 MHz
	  • 3 = 4 MHz
	  • 4 = 5 MHz
	  • 5 = 6 MHz
	  • 6 = 8 MHz
	  • 7 = 10 MHz
	  • 8 = 12 MHz
	  • 9 = 14 MHz
	  • 10 = 16 MHz
	  • 11 = 20 MHz
	  • 12 = 24 MHz
	  • 13 = 32 MHz
	  • 14 = 40 MHz
	  • 15 = 48 MHz

53296/$D030	**Turbo Enable Bit**
	This register is only available when the Turbo Mode selector is set to 'Turbo Enable Bit'. Otherwise it simply reads $FF.
	  • bit 0 (Write):
	    0 = 1 MHz + Badlines
	    1 = Use settings from menu
	  • bit 0 (Read):
	    0 = Turbo Off
	    1 = Turbo On

53370/$D07A	**SuperCPU compatible enable/disable registers** This register is only available when the Turbo Mode selector is set to 'U64 Turbo Registers' or 'Turbo Enable Bit'. This register is **write only** • Software Speed Select - Normal More information: https://www.c64-wiki.com/wiki/SuperCPU
53371/$D07B	**SuperCPU compatible enable/disable registers** This register is only available when the Turbo Mode selector is set to 'U64 Turbo Registers' or 'Turbo Enable Bit'. This register is **write only** • Software Speed Select - Turbo (20 MHz)($079) More information: https://www.c64-wiki.com/wiki/SuperCPU
53436/$D0BC	**SuperCPU Detect ($D0BC)** • SuperCPU Mode Detect Register This register is only available when it is separately enabled. This register is **read only** More information: https://www.c64-wiki.com/wiki/SuperCPU

Applies to: Ultimate 64

) Setting are only available on the Ultimate 64, firmware >= 1.33

4.4.4 Clock settings / RTC
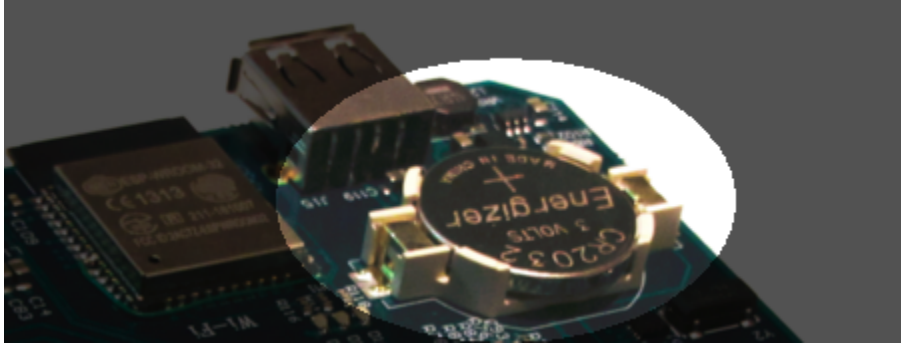
The U64 has an internal clock that is powered with a battery to keep track of the date & time.

See hardware RTC documentation

At this config screen date & time can be changed with the following options:

- Year

- Month

- Day

- Hours

- Minutes

- Seconds

- **Correction**

    – This corrects the deviation of the internal clock (RTC)

4.4.5 Setting up the MultiSID configuration on the Ultimate 64

This page explains how to use the visual SID address editor to set the UltiSID in MultiSID mode, giving you four or even eight SIDs.

In the following example, it is assumed that the settings are as follows:

- SID Socket 1 Address: Unmapped

- SID Socket 2 Address: Unmapped

- UltiSID 1 Address: $D400

- UltiSID 2 Address: $D420

- UltiSID Range Split: Off

- Auto Address Mirroring: Enabled

First step is to enter the visual editor and note that UltiSID 1 is mapped at $D400, UltiSID 2 is mapped at $D420 and auto mirroring is on, just like the settings we started with. Address mirroring means that the appearance of the available

SIDs are mirrored in the address space, such that they fill up the $D400-$D7FF range. Also pay attention that the lower line shows "Socket 1", which means that the controls (cursor and split) now have effect on socket 1.

We want to change the UltiSID settings, so we press "3" on the keyboard; the highlighted number next to the "USID1" label.

Press "M" to turn off auto mirroring, to have a better view of the settings. Note the lower line says "UltiSid 1", which we will modify.
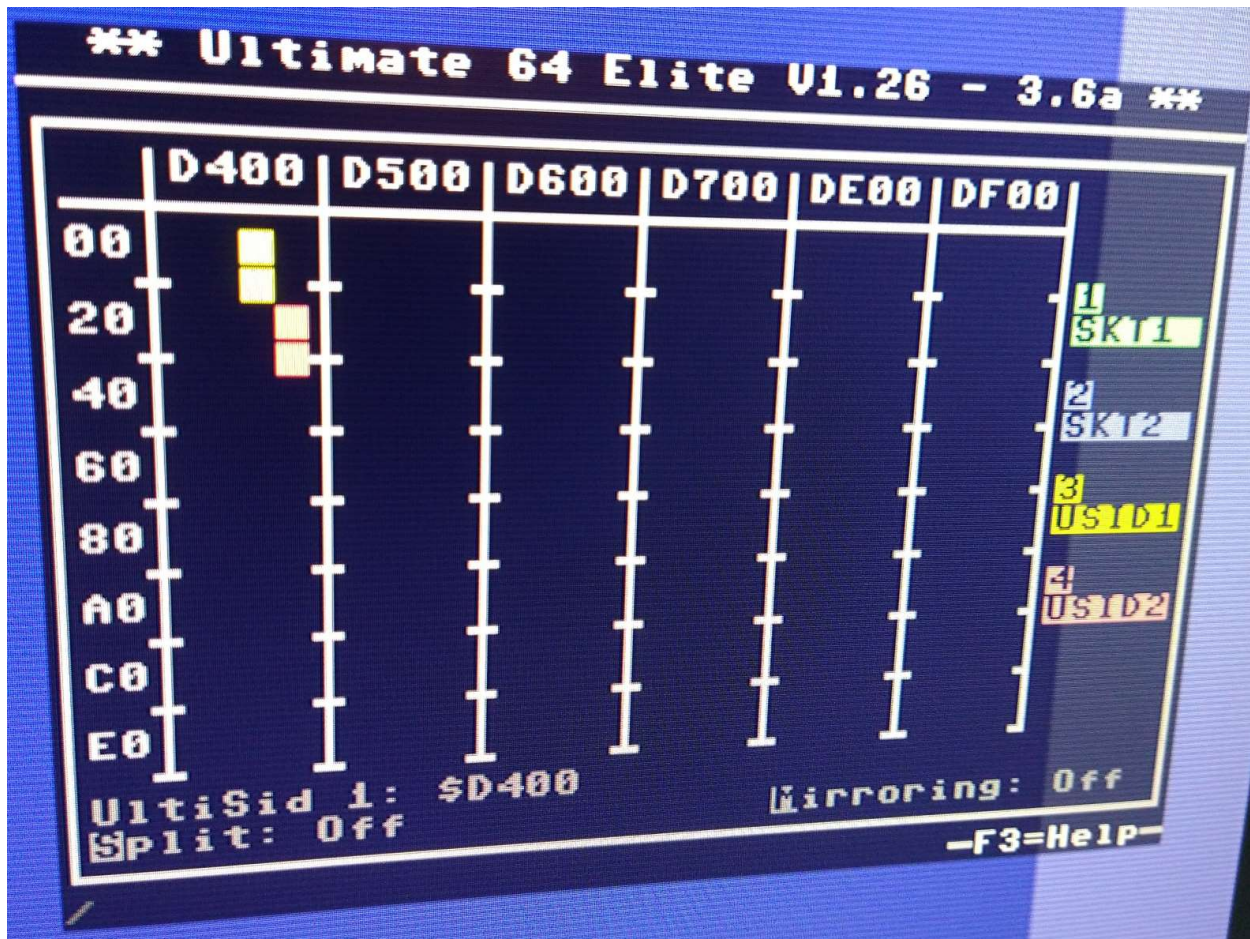
When pressing "S", the split function gets enabled. The line below shows the split is active on one address line, being A6. This means the UltiSIDs (both of them) get split in two, and the second part of it gets enabled with address line 6 set (thus $D440). Note that this instance gets a "B", indicating the second half of the split UltiSID1. The same goes for UltiSID2, which now appears at $D460.

Similarly for a split on A7; the second half appears with an offset of $80.
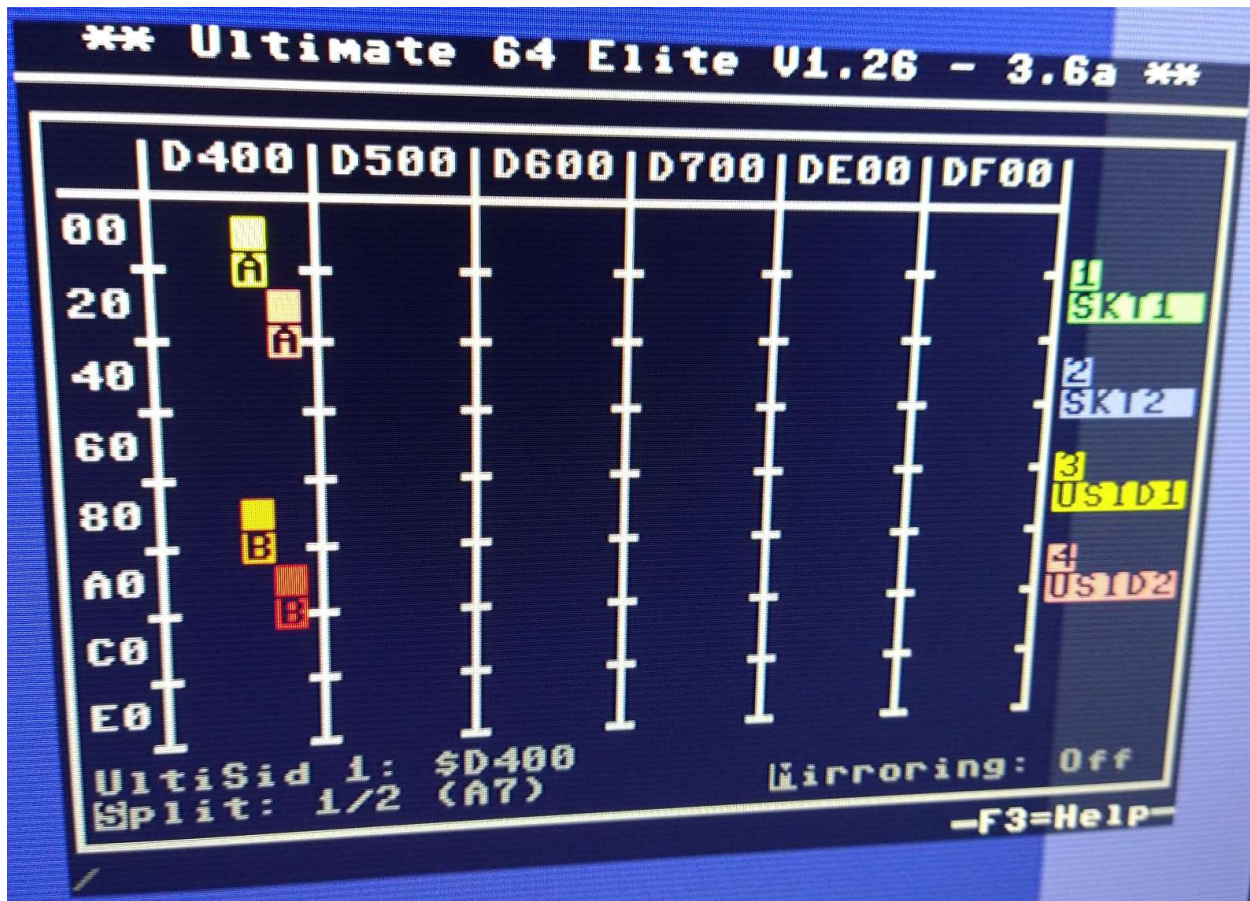
And last but not least, split on _two_ address lines, causing the UltiSID to be split into four SIDs. In this case A7 and A8, such that the second part appears at $D480 (A7=1), the third part at $D500 (A8=1), and the fourth part at $D580 (A7=1, A8=1). Also note the letters A, B, C and D.

Since the split settings are shared for both UltiSIDs, UltiSID2 also gets split in the same way. But because its base address was set at $D420 (A5=1), the splits appear also at offsets where A5=1, thus $D420, $D4A0, $D520, $D5A0. In total 8 SIDs, thus 24 voices and 8 independent filters.

4.5 Hardware

W A R N I N G ! Please always use extreme caution!

Please be extremly careful not to shorten the 5V to ground! It will burn a ferrite bead on your board! Also shorting 5V to pin 3 or any other FPGA pin will likely damage the FPGA!

4.5.1 LED Strip

Attaching a LED Strip

Using firmware version 1.10 and later, it is possible to attach a programmable LED strip to your Ultimate 64 board. The supported LED type is APA102C, with is a 5V only device with built-in PWM controller for each color, red, green and blue. At the time of writing, the firmware supports up to 64 LEDs, of which 24 are currently used. On a strip with 60 LEDs (IP30) per meter, this corresponds to 40 cm of length, which is roughly the width of the machine. Alternative are SK9822 strips.
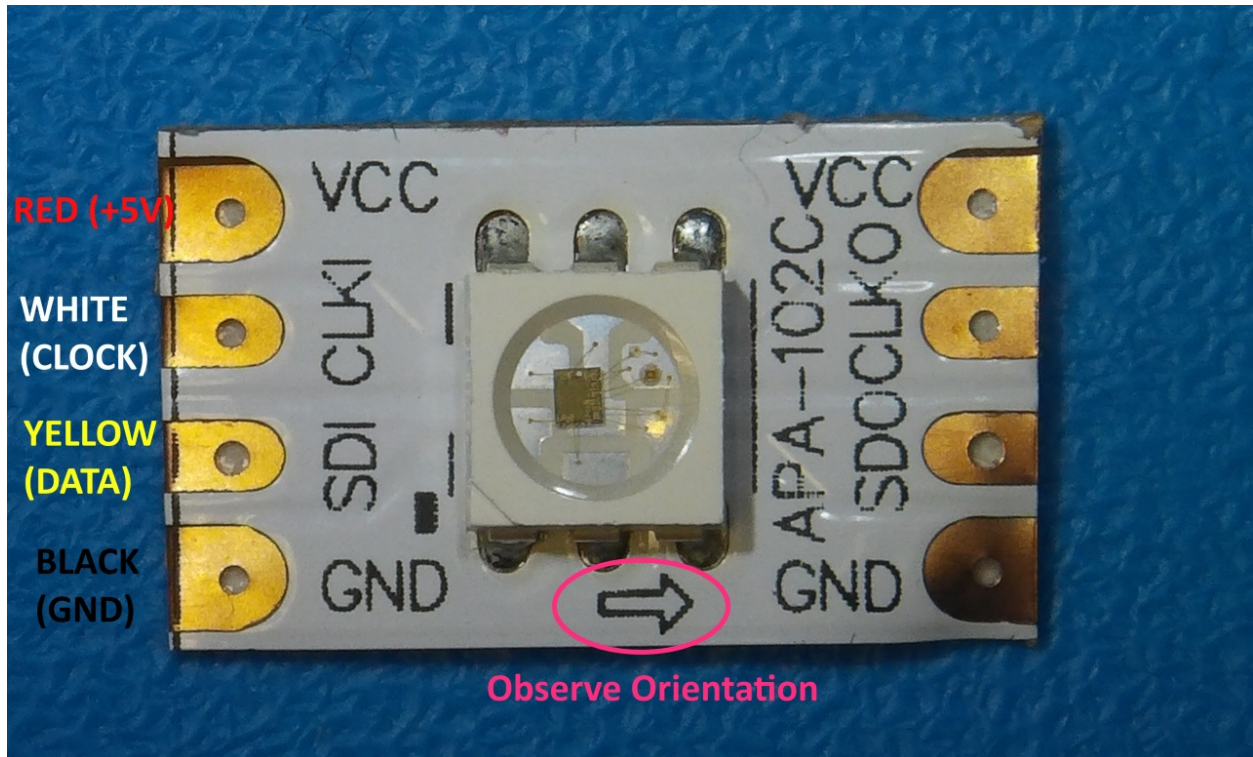
For an easy connection of the LED strip, it is recommended to connect four wires / leads to it, with a single pin socket, like the ones shown below. These wires can be obtained anywhere, and are sold -for example- as Raspberry PI jumper cables.

The following connections need to be made:

LEDstrip Pin	Name	Description	User Port
1	GND (0V)	Black wire	Pin 1, 12, or N
2	SDI	Yellow wire, Data input	Pin P
3	CLKI	White wire, Clock input	Pin 13
4	VCC (+5V)	Red wire	Pin 2

Note, the wire colors are just suggestions. Obviously it will work with other colors as well. But for the sake of not making mistakes, please at least use black and red for ground and +5V respectively.

Please refer to the drawing of the user port header, in order to see where to connect your wire leads. *USER PORT*.

LED pinheader P8

Beginning with hardware revision 1.4 you'll find the new LEDstrip pin header P8 to the left under the User Port. This can be used to attach an APA102C LED strip and simultaneously use the User Port for other purposes.

The pin header is defined as follows:

Pin Header P8:
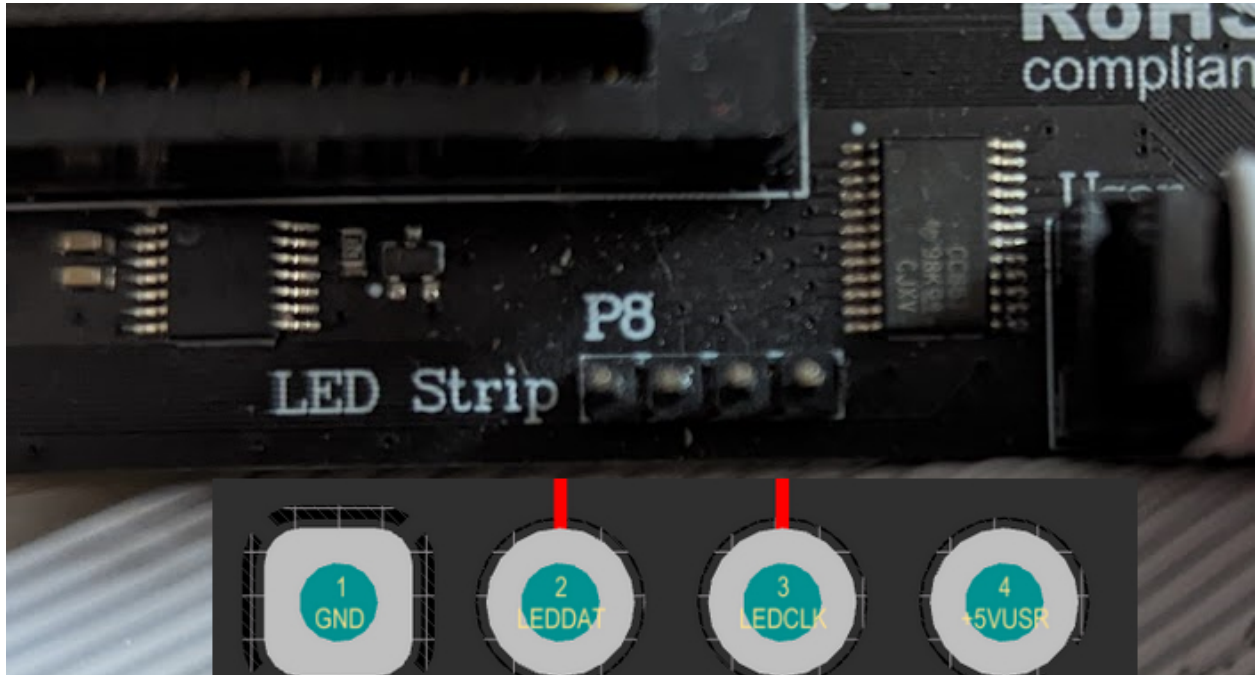
Pin 1 -> GND

Pin 2 -> DAT

Pin 3 -> CLK

Pin 4 -> +5V
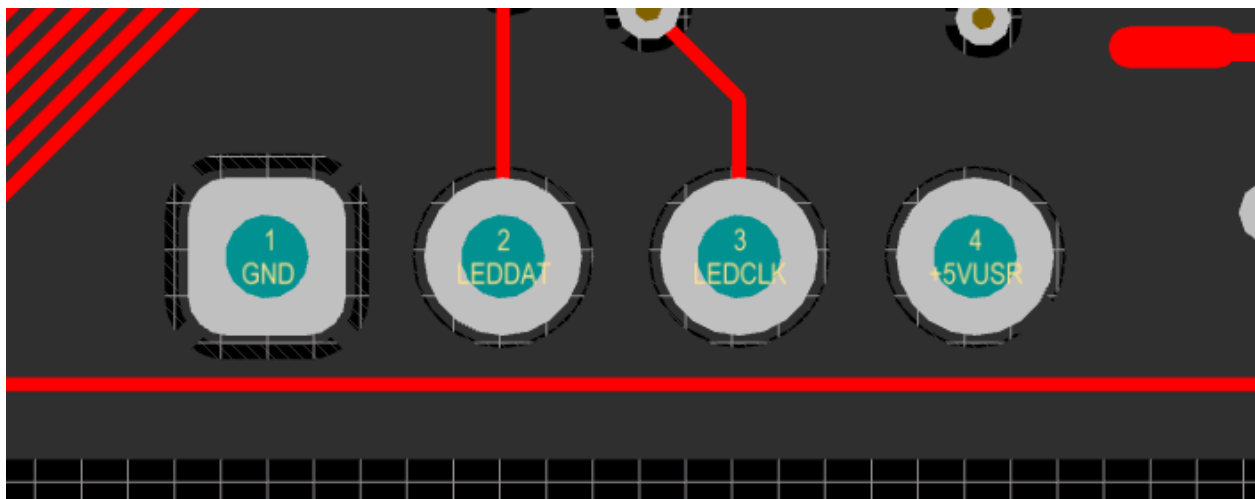
W A R N I N G ! Please always use extreme caution!

Please be extremly careful not to shorten the 5V to ground! It will burn a ferrite bead on your board! Also shorting 5V to pin 3 or any other FPGA pin will likely damage the FPGA!

4.5.2 LED Header
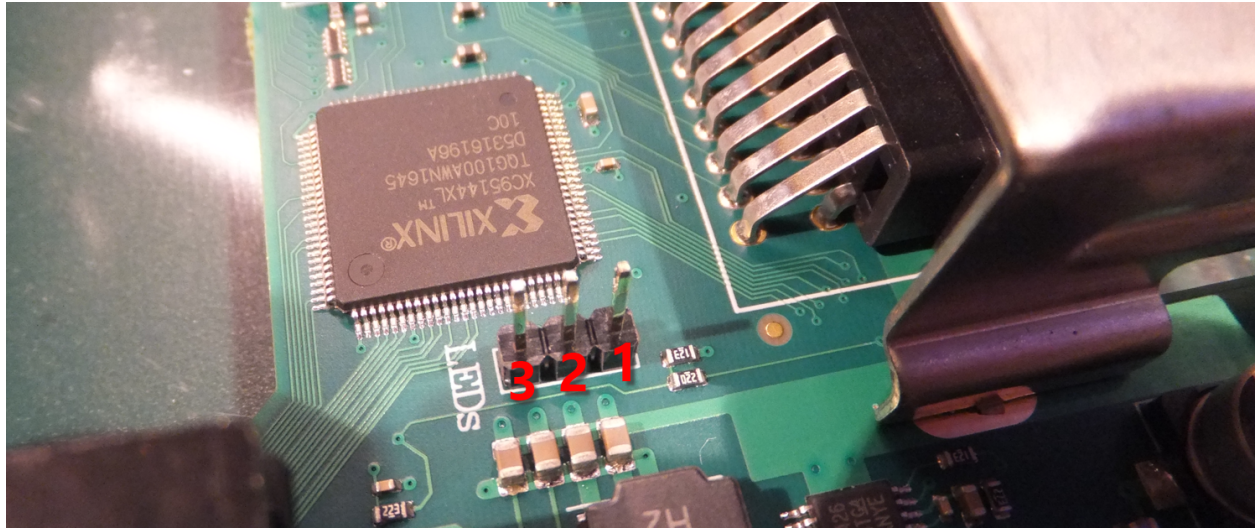
Pinout

Pin	Description
1	Power
2	Ground
3	Drive activity

With firmware 1.18 (and above) the pin assignment is not static any more.

LED options

Table 1: LED Options

Option	Description
On	LED always ON
Off	LED always OFF
Drive A Pwr	Power indicator of drive A
DrvAPwr + DrvBPwr	Power indicator of drive A and drive B combined
Drive A Act	LED ON when there is activity on drive A
DrvAAct + DrvBAct	LED ON, on activity of drive A and B combined
DrvAPwr ^ DrvAAct	Drive power LED XOR drive activity. This results in an inverse drive activity, with similar dimming as the power LED has.
USB Activity	LED ON, on USB activity
Any Activity	LED ON, on any activity (exluding tape)
!(DrvAAct)	Inverse of drive A activity, so the LED will turn OFF on drive activity instead of ON
!(DrvAAct+DrvBAct)	Inverse of drive A and B combined, so LED will turn OFF on activity instead of ON
!(USB Act)	Inverse of USB Activity, so the LED will turn OFF on USB activity instead of ON
!(Any Act)	Inverse of Any Activity, so the LED will turn OFF on any activity instead of ON
IRQ Line	LED is ON when IRQ line is high (inactive)
!(IRQ Line)	Inverse of IRQ Line, so the LED is ON when IRQ line is low (active)

With the introduction of firmware 1.18 the behaviour of the LED can be changed.

What does !(option) mean?

As we can see in the table above, there are setting options (functions) that are noted as !(function). !(function) means it is an inverse outcome of a function.

We take a closer look at the function **DrvAAct+DrvBAct**. This means when there is drive A or drive B activity (combined) the LED will turn ON. So **!(DrvAAct+DrvBAct)** will do the opposite (inverse) of the function **DrvAAct+DrvBAct**, so the LED will turn OFF when there is activity on drive A or drive B.

Example with two separate LED's

We are using a red and green LED, red will be the power LED (power) and green will be drive activity (drive). The power wire to the red led is connected to the Top pin (1) and the green led is connected to the Bottom pin (3), and of course the ground of both LED's is connected to the ground pin (2).

Because there are two separate LED's we want that the power is always ON, and the drive will only turn ON when there is activity.
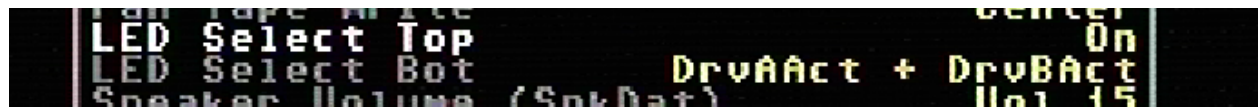
So we use the following setting:

- Power => LED Select Top: **On**
- Drive => LED Select Bottom: **DrvAAct+DrvBAct**

To swap the color assignment, just swap the settings between Top and Bottom.

Example with duo LED

We are using a duo LED that has red and green combined in one package, red will be the power LED (power) and green will be drive activity (drive). The power wire to the red led is connected to the Top pin (1) and the green led is connected to the Bottom pin (3), and of course the ground of the LED is connected to the ground pin (2).

Because there is only "one" LED package we want that power will turn OFF when there is drive activity, or else the two colors will be mixed to yellow (red+green).
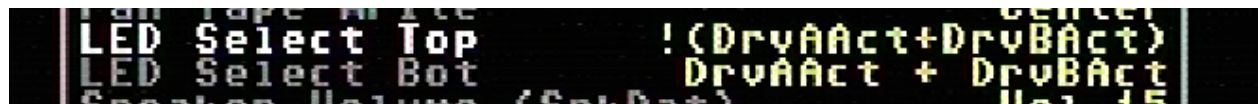
So we use the following setting:

- Power => LED Select Top: **!(DrvAAct+DrvBAct)**
- Drive => LED Select Bottom: **DrvAAct+DrvBAct**

To swap the color assignment, just swap the settings between Top and Bottom.

Technical information

Power LED

The power pin is provided with a 260KHz PWM signal, the PWM signal makes the LED less bright. The U64 has an on-board 220 ohms series resistor to lower the current, the supplied voltage 3.3V.

Drive activity LED

PWM	Description
Off	Drive Off
1/3	Drive On
2/3	Drive On + Floppy Mounted
3/3	Drive On + Floppy Mounted + Motor running

PWM = pulse width of the pwm signal.

4.5.3 Audio/Video Connector
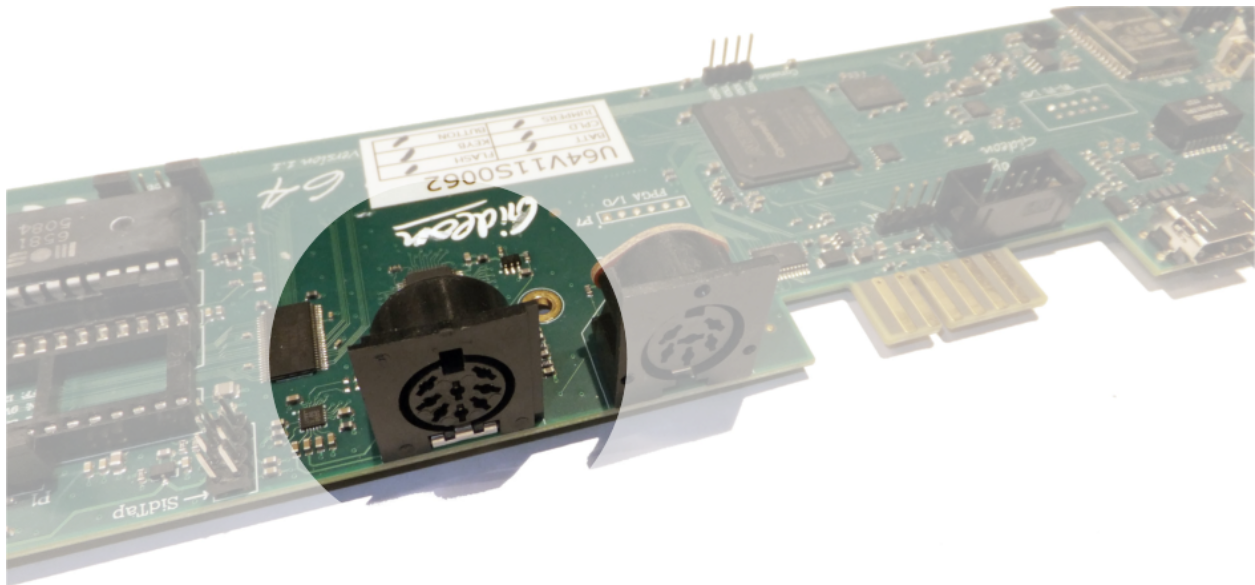
Note: The connector used is a full size 8-pin 262° DIN, also known as the 'horse-shoe' type. A 270° DIN plug does *NOT* fit without violence.

The 8-pin DIN (audio/video) connector can output several different video signals, for each type you need a special assembled cable.

Output signals

- Composite (CVBS) (PAL)

- S-Video (PAL)

- RGB (PAL)

The Composite and S-Video cables are the same as you use for the original Commodore 64.

Pinout
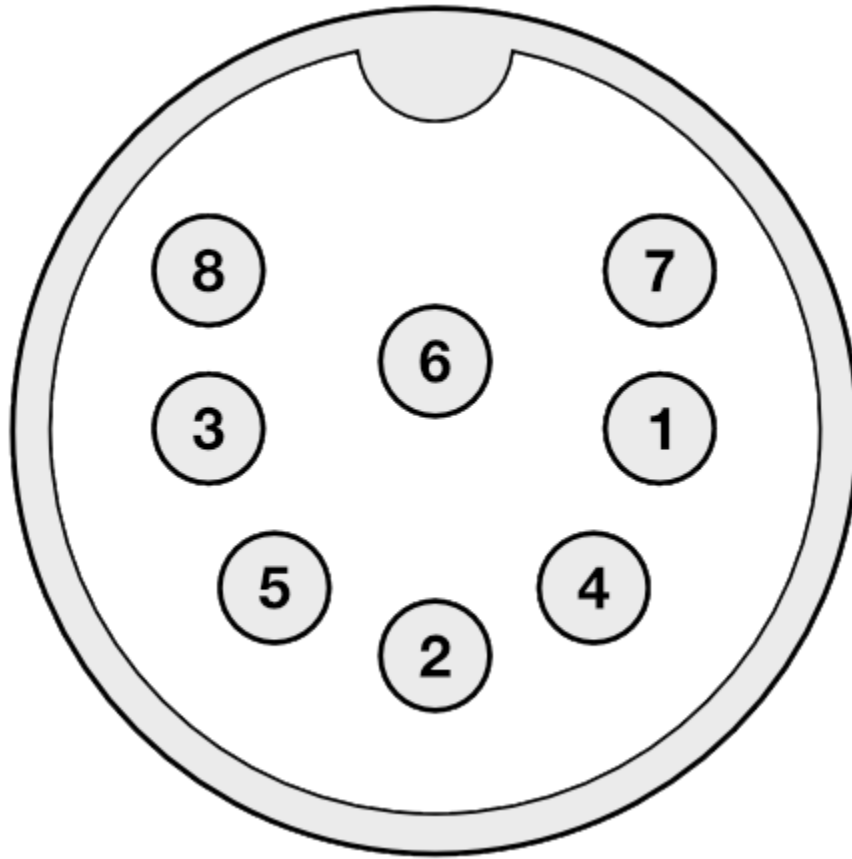
Pin	Modulated Color	RGB Mode
1	Luma (Y)	Green
2	Ground	Ground
3	Audio Out Left	Audio Out Left
4	Composite (Y+C)	Red
5	Audio Out Right	Audio Out Right
6	Chroma (C)	Blue
7	Not used	CSync
8	Not used	Fast Switching

Audio

The audio outputs have consumer electronic line levels (Line Out) and need to be connected to an external amplifier.

Video

The video signals on pin 1, 4 and 6 have a source impedance of 75 Ohms, with standard 1V-peak video levels. In the modulated color mode, Luma and Composite contain sync information (-0.3V), while Chroma does not. The video blanking of the VIC output has been corrected, and is now in accordance with specification.

The PAL output is non-interlaced, with a field rate of 50.2 Hz, and line rate of 15.639 kHz.

The Fast Switching output is a 1.5V fixed 'high' output signal (when terminated with 75 Ohms), that is used on SCART to enable RGB mode. The CSYNC output is a sync-only output that is needed on the SCART signal as a replacement of the composite video or luma signal. In RGB mode, there is *no* sync information in either R, G or B; i.e. no 'sync-on-green'.

Build your own RGB Scart cable

Pinout U64 & SCART

Connectors shown in back view
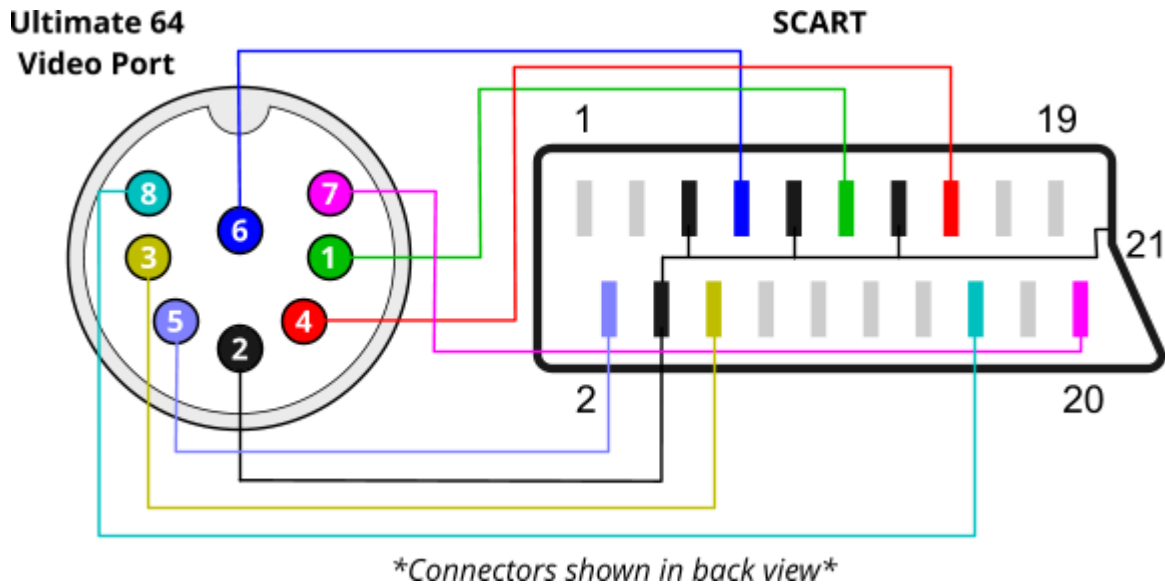
Pin	U64 video port	Pin	SCART
1	Green	11	RGB Green
2	Ground	4	and pin 5,9,13, 21(Shell)
3	Audio Out Left	6	Audio Input Left
4	Red	15	RGB Red
5	Audio Out Right	2	Audio Input Right
6	Blue	7	RGB Blue
7	Csync	20	Composite video input
8	Fast switching	16	RGB Selection, 1-3V = RGB

4.5.4 SID-TAP header
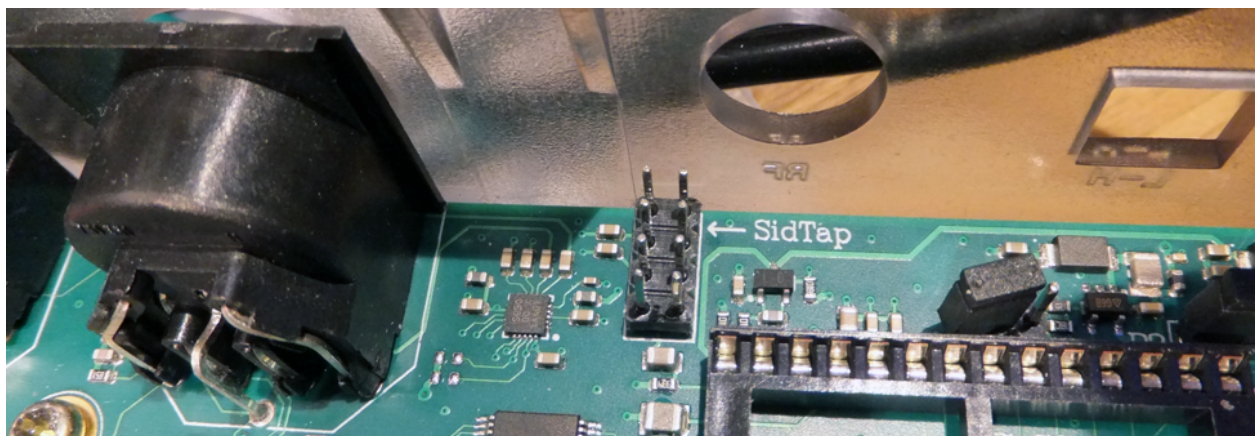
Function	Pin	Pin	Function
SID 2 OUT	1	2	Ground
Ground	3	4	SID 2 IN
Ground	5	6	SID 1 IN
SID 1 OUT	7	8	Ground
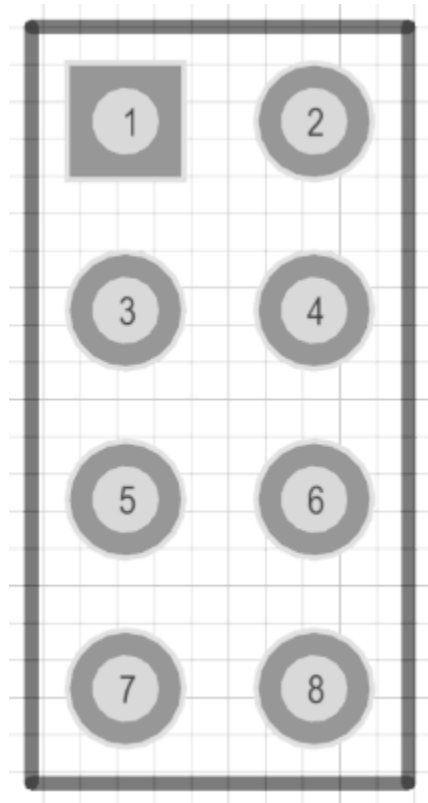
The SID-TAP header provides a direct (analog) output from the installed SID chips. This header is meant for the purists, who do not want the SID signals to be processed by the U64 board. It is possible to attach a 3.5mm jack plug directly to the SID OUT signals (left column of the header) and use it as a line-out source.

The SID inputs are not often used. However, if you like to use them, you can use the same trick, and attach a 3.5mm jack plug to the right side of the header. This connection provides a line-in connection to your SIDs. This has *not* been functionally verified!

Note that the arrow on the board points at pin 2. In other words, the image and table above are the top view of the header.

4.5.5 HDMI port
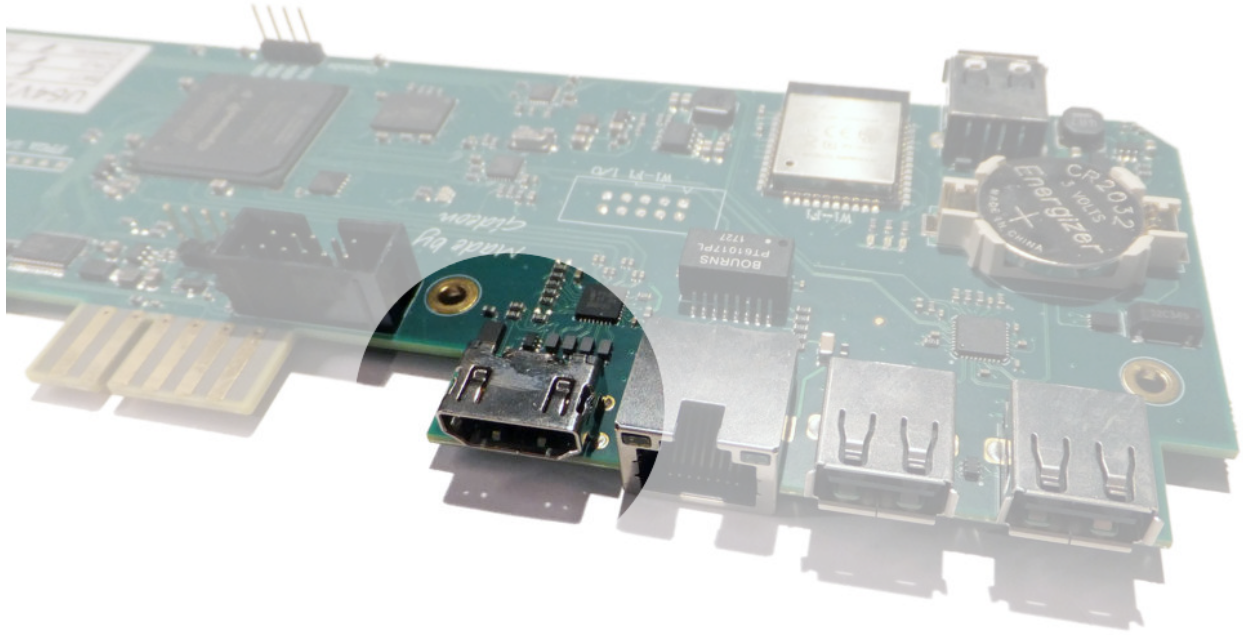
The HDMI port can be connected to a modern display (eg. television), so you can use the Ultimate-64 in the future.

Resolution of the HDMI output

The output resolution is 720 x 576 at 50 Hz (for the PAL version), also noted as 576p50. This is a standard resolution that all HDMI capable screens are supposed to support, according to the CEA-861 standard. However, to meet the exact timing of a C64, the timing parameters of this video mode are slightly modified. I have not come across screens that do not accept the generated HDMI signal.

Frame delays of the digital HDMI port

None. There is no frame buffer, so there is no need to worry. Some screens, tho, especially non-gaming screens may introduce some latency. I observed this with a 4K screen that actually resampled the output to just 30Hz. I am pretty sure displays do not have this problem. Because there is no frame buffer, scrollers and such are perfectly smooth. If the TV renders the image at the speed it is sent over the HDMI link everything will be smooth. If HDMI is used for output, there is also output generated on the PAL output (display connector).

In the configuration menu you can change several setting regarding this port.

- **U64 Specific Settings**
    - HDMI Scan lines
- **User Interface Settings**
    - Interface Type (overlay mode)

4.5.6 USER PORT
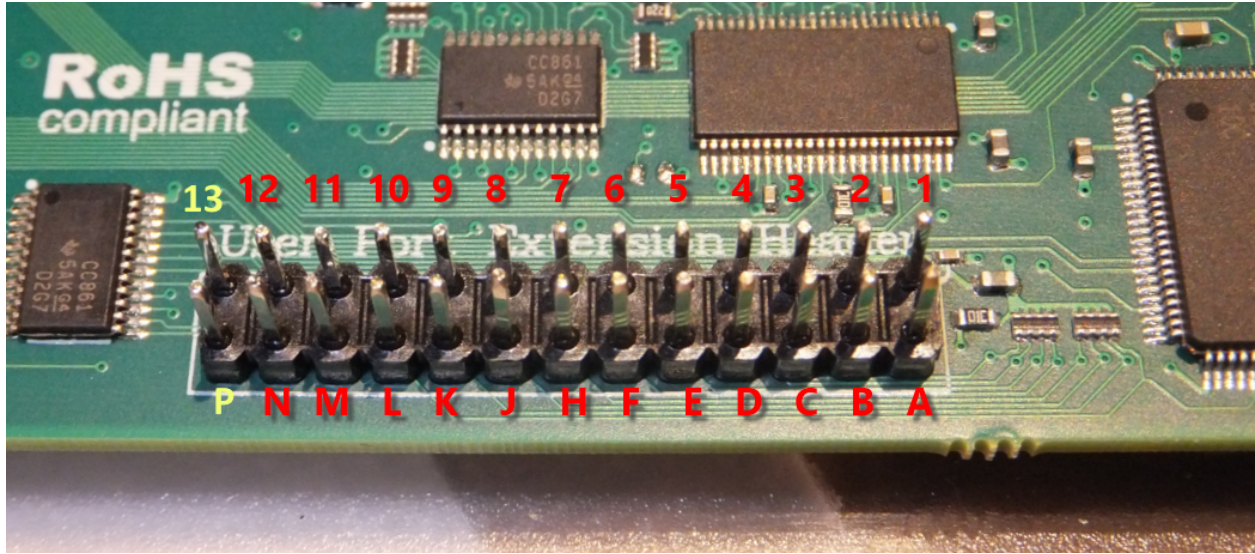
User Port Extension Header

Pin	Name	Description	Special Purpose
1	GND	Ground	
2	+5V	+5 VDC (200 mA max)	
3	/RESET	Reset, will force a Cold Start. Also a reset output for devices.	
4	CNT1	Counter 1, from CIA #1	
5	SP1	Serial Port 1, from CIA #1	
6	CNT2	Counter 2, from CIA #2	
7	SP2	Serial Port 2, from CIA #2	
8	/PC2	Handshaking line, from CIA #2	
9	ATN	Serial Attention	
10	12VDC	12VDC (Max 500mA)	
11	12VDC	12VDC (Max 500mA)	
12	GND	Ground	
13	USRCTL0	FPGA Pin, *NOT* 5V tolerant!	LED Clock
A	GND	Ground (RS232: GND)	
B	/FLAG2	Flag 2 (RS232: RxD=Both B+C)	
C	PB0	Data 0 (RS232: RxD=Both B+C)	
D	PB1	Data 1 (RS232: RTS)	
E	PB2	Data 2 (RS232: DTR)	
F	PB3	Data 3 (RS232: RI)	
H	PB4	Data 4 (RS232: DCD)	
J	PB5	Data 5	
K	PB6	Data 6 (RS232: CTS)	
L	PB7	Data 7 (RS232: DSR)	
M	PA2	PA2 (RS232: TxD)	
N	GND	Ground (RS232: GND)	
P	USRCTL1	FPGA Pin, *NOT* 5V tolerant!	LED Data

Applies to: Ultimate 64

Note that the row 1-12 corresponds to the upper row of the User Port card edge fingers on a C64, and that the row A-N

---

corresponds to the bottom row of the User Port card edge fingers.

Pin 13 and P, the two left most pins on the header, are added for add features to the user port. One of the intended usages of these two pins is the generation of the "9 V~" on the user port extension board.

There is one other use of pin 13 and P, which is the control of a LED Strip. In this case, pin 13 the LED Clock and pin P is the LED Data, assuming a LED strip that is based on APA102C.

In the LED Strip configuration menu, the use of pind 13 and P can be chosen. When the LED Strip is set to 'off', the 9 V~ generation is enabled.

Attaching the User Port Adapter

The user port adapter accessory should be connected using a 26-pin ribbon cable. Note that there is only ONE correct way to connect it, and other ways may result in damage of the U64 itself or to the accessory.
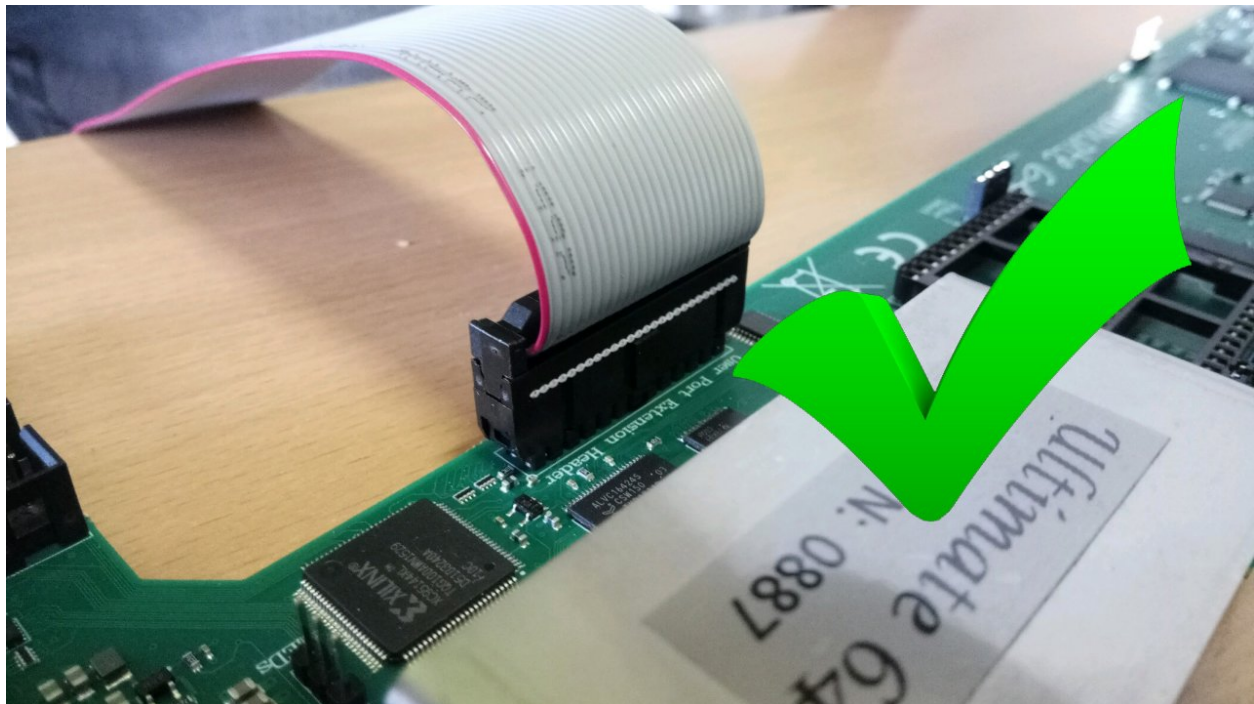
Pay attention to the orientation!! The red wire should line up with pin 1!!!

Please make sure the connector is not shifted a whole row, or to the left or to the right:

4.5.7 Ethernet port
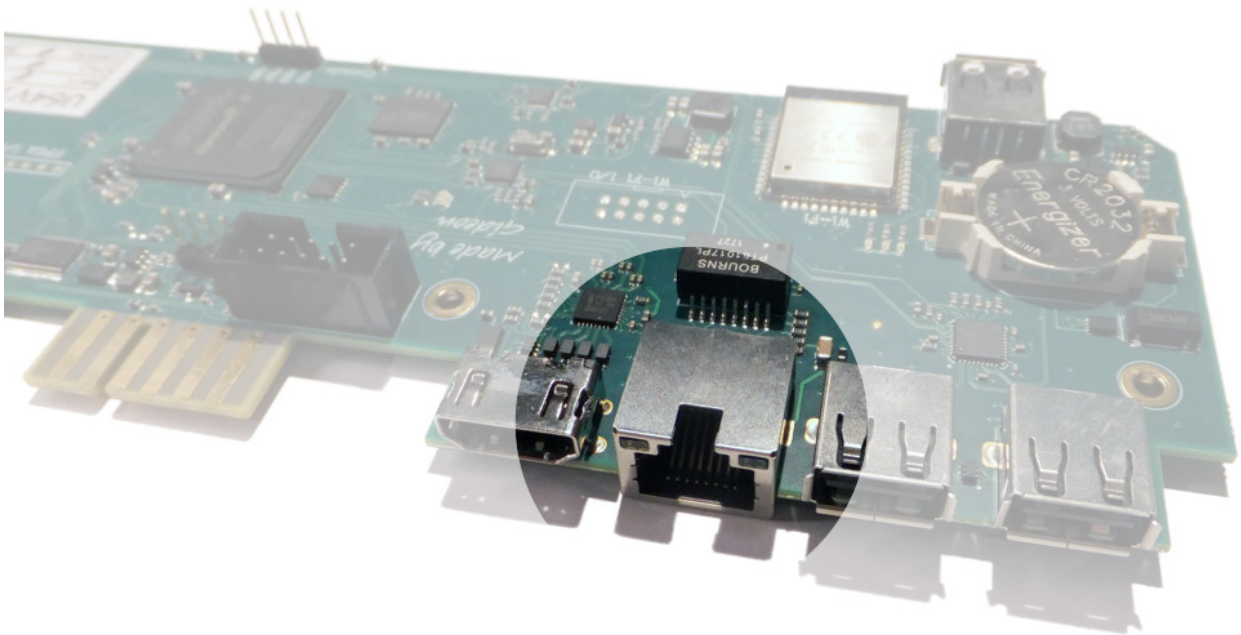
The ethernet port can be connected to a switch to communicate with your Ultimate over your own local network.

The Ultimate 1541-II+ and Ultimate 64 have an integrated ethernet port.

Ultimate 1541-II and compatible usb ethernet adaptor

The Ultimate 1541-II needs a compatible USB2LAN Ethernet adaptor that is connected to the USB port of the Ultimate 1541-II. Any USB2LAN adapter that uses an AX88772/-A/-B chip should work. On firmware 3.2 and 3.3 the ethernet adaptor is unstable. This has been fixed in firmware 3.4.
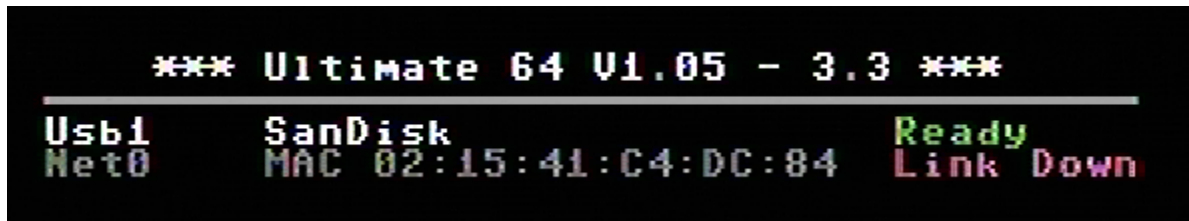
If your Ultimate product has an ethernet port, you can see this when entering the Ultimate menu. The picture above shows a **Net0** port, this indicates ethernet is accessible.

Network settings

Within the network settings menu the ethernet port can be configured as needed, so it matches with your local network settings.

The Network settings menu is only visible if a ethernet port is available.
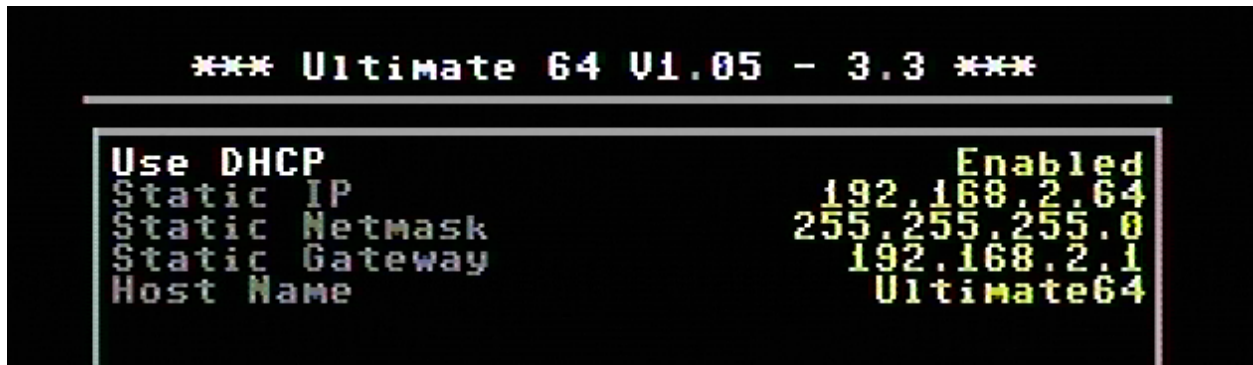
Menu name	Explanation	Options (bold default)
Use DHCP	Get IP settings from local DHCP Server on your network. Most home routers provide these settings by DHCP. If this is not the case please consult your internet/network provider.	Disabled **Enabled**
Static IP	Set custom IP settings, DHCP has to be disabled. *This field can be left empty if DHCP is enabled*.	Inputbox
Static Network-mask	Set custom networkmask, DHCP has to be disabled. *This field can be left empty if DHCP is enabled*.	Inputbox
Static Gateway	Set custom Gateway, DHCP has to be disabled. *This field can be left empty if DHCP is enabled*.	Inputbox
Host-name	Set unique network hostname to identify your Ultimate product.	Inputbox

RR-net compatibility

Some of you may be familiar with the RR-net solution that brings Ethernet to the C-64. Currently, the built-in Ethernet port does not provide RR-net compatibility.

The Ethernet port is used natively by the firmware

There is some primitive support for file-transfer using FTP (needs improvement), and it is possible to connect to the Ultimate using a VT-100 terminal program on the Telnet port (port 23). This gives the possibility to control the machine remotely, and swap disks without actually interrupting the program running on the C-64.

Ultimate-related BBS and IRC clients

Since RR-net compatibility is not yet provided, some specific Ultimate software has been developed for accessing BBS and IRC networks. Ulimate Lib is a library (written by **Scott Hutter**) that gives easy access to the full TCP/IP stack that the Ultimate provides. This library comes along with two sample applications: Ulimate Chat and Ulimate Term, this one co-developed with **Francesco Sblendorio**.

Applies to: Ultimate 1541-II, Ultimate 1541-II+, Ultimate 64

4.5.8 USB ports
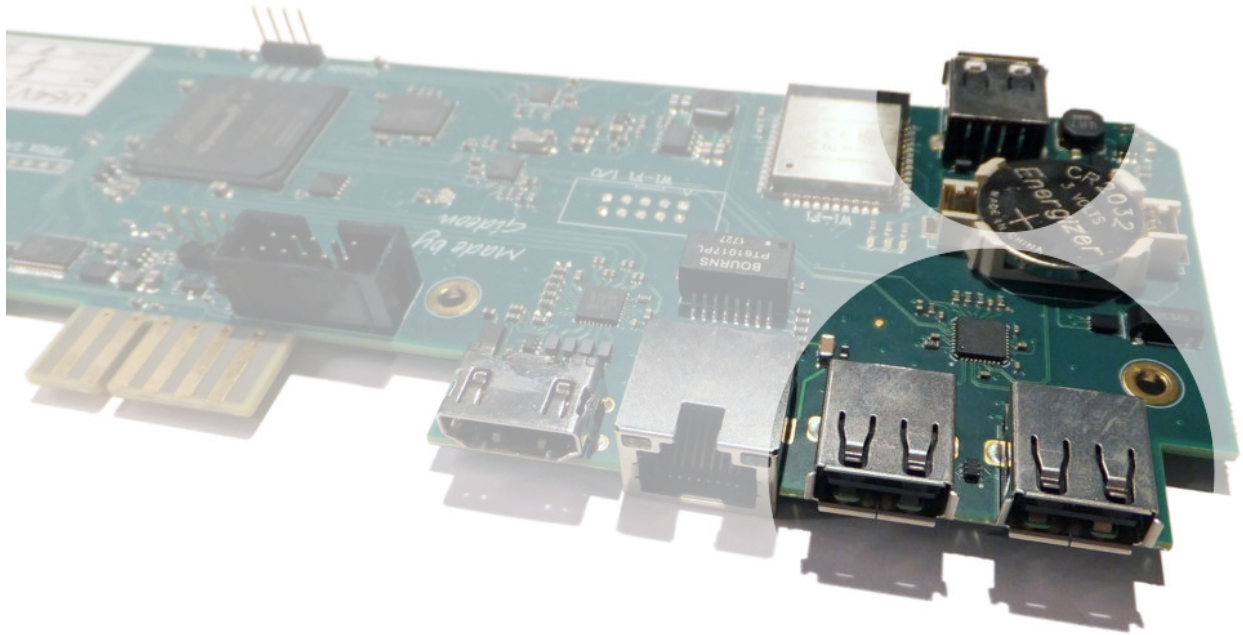
The Ultimate-64 supports most USB sticks and Flash card readers out of the box. It also supports USB 2.0 Hubs. It is recommended to use powered USB hubs only (with an external power supply).

USB 1.1 Hubs are not supported.

There are three USB 2.0 (**no USB 3!!**) ports available on the board; two on the back and one on the "internal" side of the board/case.

Usb0	External USB (next to NIC)
Usb1	External USB (edge of PCB)
Usb2	Internal USB

4.5.9 Joystick Ports

Pinout

Pin	Description
1	Up
2	Down
3	Left
4	Right
5	POT Y
6	Fire
7	+5V
8	GND (0V)
9	POT X

–> Although the U64 can handle 5V signals on the joystick port, it is *NOT RECOMMENDED* to pull any digital pin (up, down. . .) to +5V directly. Be careful with some Sega controllers that do this. It might cause latchup in the Xilinx

PLD, or a filter component to break.

Joystick Swap

With the introduction of the Elite boards, the pins are no longer directly attached to the same 6526 CIA pins as the keyboard, like on a C64. Instead, these signals are combined in the Xilinx PLD. This makes it possible to switch the joystick ports from software.

To switch the joystick ports, enter the menu and press C= + J.

Paddle Support

The POT-X and POT-Y pins support reading analog values from a controller, like a paddle. The original paddles are simply potentiometers with a knob and can function as a simple control for a race game steering wheel or paddle control in Pong.

On the original C64, the paddle ports are sampled by the SID chip. Because the Ultimate 64 does not require a real SID chip to operate, the POT-X and POT-Y signals are handled locally on the board itself and are therefore NOT connected to the SID sockets.

How does the paddle sampling work?

This aforementioned potentiometer (variable resistor) of a paddle controller connects between the +5V and the POT-X or POT-Y input pins. This external resistor charges a capacitor. This capacitor is discharged by the SID for 256 machine cycles and is allowed to charge during another 256 cycles by the external resistor. Once the voltage on the POT-X pin crosses a certain threshold, the counter value is latched. Then, at the end of the 256 'charge cycles', the capacitor is again discharged for 256 cycles. This delivers a readout value every 512 machine cycles (PHI2) (~2 kSPS).

Because of a large leakage current in the original SID from these pins, the voltage observed on the POT pins raises relatively fast for the first volt or so. Above this voltage, the potentiometer becomes dominant in charging the capacitor. The latch-threshold is about 2.25V.

KoalaPad on the U64

On te U64, the charge on the capacitor depends predominantly on the external resistor in the paddle. To get a reasonable match of the readout values, a lower threshold was chosen, a bit less than 1.6V. However, this gives issues with external devices like Koalapad, because it uses the POT pins differently. It does not use a resistor to allow a capacitor to charge, but it makes use of how the conversion works. Koalapad lets the POT-X and POT-Y pins sit just below the threshold (of a SID), and pulls it over the threshold when it 'knows' that the counter in the SID contains the intended value to send, depending on where the drawing pen presses the pad.

Because the threshold in the U64 is much lower, it does not read the Koalapad position but it just reads a low value instead; the comparator trips too early; way before the Koalapad tries to pull the voltage over the threshold.

In order to fix this, some resistors need to be changed on the U64 board near the keyboard connector, on the very lower right (before serial number ~3000):
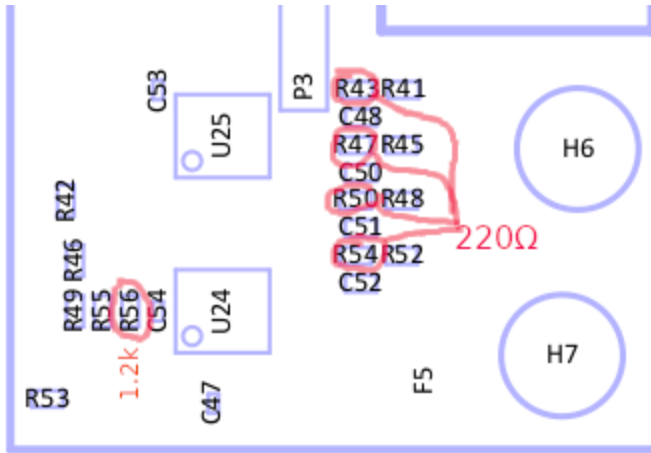
This patch causes a better discharge during the discharge phase (R43, R47, R50, R54), and raises the threshold to almost 1.8V (R56). If Koalapad still doesn't work for you with this patch, you can experiment by increasing R56 to 1.5kΩ even, setting the threshold at 2.0V. Just be aware that real paddles will have a smaller active range, as the port will read 255 sooner, with lower resistor values.

Newer U64 boards have the threshold set to 1.85V, which provides a safe margin for the Koalapad (R55 = 5.6 kΩ, R56 = 3.3 kΩ).

4.6 Data Streams

4.6.1 Introduction

Context

The Ultimate-64 is capable of streaming real time data over its built-in Ethernet port. Currently, there are two different types of streams that the machine can output:

- VIC Video Stream

- Audio Stream

- Debug Stream

This page describes these features in more detail.

Goal

This feature is intended for debugging, regression testing and analysis. However, there are more uses for it; such as making pixel-perfect recordings of a demo, or a digital audio recording without conversion to analog first and sampling it on a PC. Or, even a real time share over the internet to a friend!

4.6.2 Networking Background

The network streams are based on UDP. This means that they are connection-less, and can be captured off the network by the addressed node. UDP does not guarantee delivery, so it may be possible that packets get lost. However, on a local network this usually does not happen.

Aside from its simplicity, UDP has another advantage: it can be used to send data to multiple receivers at once. This is called 'multicast'. The Ultimate is capable of sending the datastreams in unicast mode (to one specific IP address), in multicast mode (to a multicast IP address), or in broadcast mode. When the stream is transmitted in multicast mode, the receiver needs to 'join' the multicast group. Note that this will only work when the switch supports IGMP snooping, otherwise the switch will broadcast the data over all of its network ports; thus treating the packets as broadcast packets.

When the switch supports IGMP snooping, it will 'see' the network packet that is generated by the client who wants to receive the stream when it joins the multicast group. Such a request is sent using the IGMP protocol. Once the switch 'sees' this packet, it will enable the transmission of the requested stream to the physical port on the switch that requested the data (the port the PC is connected to). No other ports will receive the data, so the network is not flooded.

Unicast Example

When your switch does not support IGMP snooping, the use of unicast is recommended. Especially when there are other users of your local network, that are behind WiFi links, it is not recommended to flood the network and get angry looks from everyone else in your household.

Unicast is the simplest form. Let's say your PC is at 192.168.0.100; either statically configured or obtained by DHCP, and your Ultimate has the IP address 192.168.0.64; again either statically configured or obtained from the DHCP server. You can then simply instruct the ultimate to stream the data to IP address 192.168.0.100.

The receiving software will open a socket, listening to a specific UDP port number. It is necessary to send the stream to this specific UDP port number, otherwise the stream cannot be received.

Multicast Example

When using multicast, a specific range of IP addresses is used for transmission: 224.0.0.0 to 239.255.255.255. Some of these addresses are reserved for a specific type of communication, like IGMP, MDNS, and such.

Let's say the Ultimate is configured to send the stream to 239.0.1.64. This is in the free range for multicast on the local network (LAN). The PC application starts by telling the network infrastructure and its own routing tables that it wants to listen to 239.0.1.64. From that moment on, the stream is 'seen' by the PC and the application receives the data. Once the PC application is done, it 'leaves' the multicast group, and the PC ceases to receive the data.

Note that the IGMP switching is based on IP address. Therefore it is not possible to use the UDP port number to differentiate which stream to receive and which not. Use different multicast IP addresses if you like to receive more than one stream.

4.6.3 Starting and Stopping Streams

The available streams can be started and stopped in different ways. When the machine is turned on, it does not transmit any stream.

Using the menu

A stream can be started and stopped from the "action menu", which is brought up by pressing 'F5'. The start/stop options appear in that menu. When selected, the Ultimate will ask for the IP address to send the stream to. The default IP address is configured in the configuration screen "Data Streams". Note that changing the IP address in the configuration screen *does not* start or stop a stream; it is only the place where the last entered data is kept for later reuse.

The format of the destination is either an IP address, or a valid DNS name. The port number can be given as well, by adding it after a colon (:).

Examples:

Destination string	Meaning
192.168.0.119:11000	unicast address on the local network, port number 11000
myserver.com	unicast address, using DNS and default port number
myserver.com:4567	unicast address, using DNS and specific port number
239.0.1.64	multicast address, using default port number
239.0.2.77:64738	multicast address with port number specified

Using the TCP command interface

Using the TCP socket '64', the following commands have been added to start and stop streams.

- Enable output stream: FF2n
- Disable output stream: FF3n

whereby 'n' is the stream number (see below)

The generic command structure is:

- <command word, little endian>
- <command parameter length, little endian>
- <command parameters>.

The Disable command does not take any parameters. The enable command can have two parameters; neither of them are required.

- Parameter 1: Duration of the stream enable (0 = forever)
- Parameter 2: Destination (string).

The duration is given in system ticks, which come at an interval of 5 ms.

So to elaborate on the commands:

To enable stream 0 for one second (200 ticks), to the pre-configured destination address, the bytes sent over the TCP control port are:

`20 FF 02 00 00 C8`

To enable stream 0 indefinitely to '192.168.0.119', you'd send:

`20 FF 0F 00 00 00 192.168.0.119` (the latter part being ASCII)

-> Command length = string length + 2 bytes.

To disable stream 0, you send:

`30 FF 00 00` (length of parameters is 0)

4.6.4 Available streams

VIC Video Stream (ID 0)

The VIC Video Stream is the stream that presents the active part of the video output of the VIC. The ID of this stream is 0.

Each UDP datagram contains a header and pixel data. The header is as follows:

- Sequence number (16 bit LE)
- Frame number (16 bit LE)
- Line number (16 bit LE). If this is the last packet of the frame, bit 15 will be set.
- Pixels per line (16 bit LE), always 384
- Lines per packet (8 bit), always 4
- Bits per pixel (8 bit), always 4
- Encoding type (16 bit), always 0 for now. May be '1' later for RLE encoded data.

Example packet:

The following packet contains data of the third frame, line 100 - 103, starting with dark blue:

```
A1 00 02 00 64 00 80 01 04 04 00 00 66 66 66 66 66 66 66 ...
seq #|frm #|line#|width|lp|bp|-enc-|data...................
```

So in total, the pixel data is preceded with 12 bytes of header data. The header is followed by pixel data, coded in 4-bit VIC colors, little endian, thus nibble 3..0 first. 4 lines of 384 pixels, equals 768 bytes of pixel data. Total UDP datagram = 780 bytes.

To capture an entire frame, you first wait for a packet with bit 15 set of the line number, then capture until you get another packet with bit 15 set of the line number. You should then have 68 packets of 4 lines (totaling 272 lines). Note that the video is cropped to 384 x 272, and aligned with the reference images of the VICE test suite. The actual output to the screen is 400 x 288 in PAL mode.

In NTSC mode, the output stream carries 384 x 240 pixels. The actual output to the screen is 400 x 240.

Audio Stream (ID 1)

The Audio stream is taken from the output of the audio mixer. Thus, the stream received over the network contains the same data as the data sent over HDMI and back to the audio codec for conversion to the analog signal that is available on the DIN connector. The ID of this stream is 1.

The format of the audio stream is simpler than the video stream. The only two bytes that are sent as a header in front of the raw audio data is the sequence number of the packet, since the stream was enabled. This allows for detection of missing packets.

The sequence number is followed by 192 stereo samples in 16-bit signed, little endian format; left and right interleaved, starting with the left channel. Thus, the total UDP packet size is 770 bytes: 2 header bytes, and 192 times 4 bytes per sample:

```
00 00 fe ff 02 00 ff ff 03 00 fd ff 00 00 ...
-seq- left--right-left--right-left--right-...

01 00 ff ff 03 00 fd ff 00 00 03 00 ff ff ...
-seq- left--right-left--right-left--right-...

02 00 fd ff 03 00 fd ff 00 00 fe ff 02 00 ...
-seq- left--right-left--right-left--right-...
```

The sample rate of the stream is close to 48000 Hz. It depends on the video mode. The actual sample rate for PAL is:

(Fc * 16/9 * 15 / 77 / 32) = 47983 Hz (Fc = 4433618.75 Hz) (48kHz: -356 ppm)

For NTSC, the audio clock is derived as follows:

(Fc * 16/7 * 15 / 80 / 32) = 47940 Hz (Fc = 3579545.45 Hz) (48kHz: -1243 ppm)

Debug Stream (ID 2)

This is an advanced feature

The Debug stream is new in version 3.7 of the firmware (V1.28 and higher). It streams the CPU, VIC or 1541 CPU accesses directly to the network. What does this mean? It means that the program flow and memory accesses can be traced *real time* from the machine. This essentially opens up the possiblity to trace program flow and timing, down to clock cycle accuracy. In addition, by adding the CPU access of 1541 Drive A to the possible output streams, interaction between the host machine and the drive can be analyzed.

Due to the bandwidth limitation of the 100 Mbps network port, it cannot be used in combination with the Video Stream (ID 0), nor can all three stream sources be used at the same time. Each stream occupies roughly 32 Mbps!

The following modes are supported:

- 6510 Only

- VIC Only

- 6510 & VIC

- 1541 Only

- 6510 & 1541

Each cycle / memory access occupies one 32-bit word in the output stream. The format of the 6510 and VIC streams are equal, with the exception of bit 31, which literally indicates the state of the PHI2 signal during the access. Because VIC accesses can also occur when PHI2 = 1 (Bad lines, for instance), the distinction whether a cycle is a CPU or a VIC cycle is made based on AEC and PHI2.

The 1541 CPU cycle also occupies one 32-bit word, but the data format is slightly different. The 1541 stream also contains the state of the signals ATN, CLOCK and DATA from the IEC bus, such that the data flow over this cable can be seen as well.

The first 32-bit word consists of a 16-bit sequence number and a reserved 16-bit word. Following this first 32-bit word, 360 data entries follow, each 32-bit. Hence, the payload size of the datagrams is 1444 bytes:

```
05 16 00 00 xx xx xx xx yy yy yy yy ...
-seq- resvd |- entry 0 -|- entry 1 -|
```

The format of the data entries are as follows:

Table 2: Debug Stream Format

bit	31	30	29	28	27	26	25	24	23..16	15..0
6510 / VIC	PHI2	GAME#	EXROM#	BA	IRQ#	ROM#	NMI#	R/W#	Data	Address
1541	'0'	ATN	DATA	CLOCK	SYNC	BYTE_READY	IRQ#	R/W#	Data	Address

One way to visualize this stream, is to capture it to a file using a python script, such as found in the repository: grab_debug.py

There is a preliminary tool to 'decode' the binary file that is captured. In the tools directory there is a 'dump_bus_trace.c', which is compiled by running 'make' in that directory. Dump_bus_trace converts the captured stream to a *vcd* file, or value change dump file, which can be viewed with GtkWave or any other software tool that can visualize it. This is all very preliminary. A lot of work needs to be done to find and visualize exactly that what you're looking for, since the files are very large. E.g. a disassembly of the captured CPU bus may come in very handy. It would be better to rewrite the messy 'C' tool as a set of Python scripts.

4.6.5 Viewing and recording

Windows

Thanks to Martijn Wieland (TSB), there is a viewer for the stream for Windows. This tool can be obtained here: .

Linux/Mac

Jimmy (DusteDdk) made a Linux viewer, more info check:

This version can also be compiled on Mac OS.

Python scripts

Also there are example scripts in the 'python' directory of the ultimate repository to show how grabbing of video data and audio data is done. This example scripts can be seen here: grab.py and grab_audio.py

4.7 Ultimate SID player auto configuration

What this does, is to select the correct socket when the internal player is started. The autoconfig selects the socket based on what type of SID is found in the socket and for what type of SID the tune was written. If the type of SID is not found, it may revert to UltiSID. This means that it will look for a 6581 if the SID header indicates this. This may lead to confusing results if the SID type installed is 8580.

If the SID types are correct, it will set the address(es) correctly, as well as the panning:

- Single SID tunes will pan to center.
- 2SID (Stereo SID) will pan to L3/R3.
- 3SID will pan to L4/Center/R4.

So it is a best-effort algorithm that works best for single SID tunes and you have one 6581 and one 8580 installed.

The SID autoconfig temporarily overrules the user configured audio mixer settings with the automatic configuration!

4.8 How to manuals

4.8.1 Using a real Datasette

The Ultimate64 can make use of a real Datasette the same way a real C64 can. Just connect it to the tape port and apply the usual commands like LOAD, LOAD"" or shift-run/stop.

Wrong kernal

Please note that the wrong kernal, or an activated cartridge probably will cause errors.

If you get the "missing file name error" it will probably be caused by a wrong kernal. If you have doubts if you have flashed the wrong kernal it is recommended that you copy the kernal used by Vice and flash that as the kernal file on your U64.

4.8.2 Creating own .tap files

The Ultimate64 is able to make a .tap backup of your original C64 tapes. The U64 will use these .tap files to "emulate" the datasette.

1. Open the file browser and select the folder where you want to save your .tap

2. Hit F5 and select "tape"

3. Select "sample tape to tap"

4. Enter the filename you want to use and hit ENTER

5. "Start tape when back in C64 screen" hit ENTER

6. Exit the U64's menu and press the PLAY button of your Datasette

7. Wait until the tape is finished

8. Go back into the U64's menu and hit F5

9. Select "Tape"

10. Select "Finish Rec. to Tap", your tap is complete

Tip:

If you use a USB stick with a read/write LED you can see when data from the tape is being written. This helps you to see when the data on your tape starts and when it is finished, the U64 has no kind of indicators for that.

FIVE

USING WIFI ON THE ULTIMATE 64

5.1 Instructions

Before the WiFi module on the U64 can be used, it needs to be programmed with a supplied ".esp" file. This file contains the firmware for the ESP32 module. Without this firmware, the Ultimate is not able to 'talk' with the WiFi module, and will therefore not detect it, and not be able to set up a network connection.

In order to install the firmware, simply browse to the supplied ".esp" file and select the option "Flash into ESP32". The ultimate will then start a programming cycle of the module in the application background. The user is still able to use the ultimate application while the module is being programmed. Do not attempt to access the module during this time. After about two minutes or so, a popup appears that informs you that programming the ESP32 is complete. From that moment, the WiFi line in the main screen will be operational.

5.2 Setting up a connection

Note that the ESP32 only supports WiFi on the 2.4 GHz band. Therefore, this band must be enabled in the configuration of your accesspoint.

When the WiFi line in the main screen shows a MAC address, the module has been recognized and is waiting to connect to a WiFi access point (AP). Take the following steps to connect:

- Select the context menu item "Show APs..". A list appears of access points that the module has found. The list is sorted by signal strength, the strongest signals / nearest APs on top.

- Select the network you want to connect to. A popup appears to enter the network password.

- Once you enter the password, you can leave the list of APs by pressing CRSR-LEFT.

- The WiFi should now connect and show an IP address.

5.2.1 Invisible SSIDs

When you are using an accesspoint where the SSID is not published on the network, you can also manually connect by entering the SSID and password in the configuration menu. Press F2 to enter the configuration and navigate to "WiFi Settings". There you can enter the SSID and password manually. Do not forget to also specify the type of authentication. This is required for a successful connection.

5.3 Functionality Available on WiFi

The WiFi module is used as a transparent network adapter to the Ultimate 64 management application. Therefore, the WiFi can be used for telnet, ftp, modem emulation and also to connect to the internet with the appropriate software or built-in functionality such as searching the online Assembly64 database. The WiFi *cannot* be used for streaming video and audio. This is only available on the LAN port.

5.4 Tin Foil Hats

For those that wear tin-foiled hats: When the WiFi module is disabled in the configuration, or by means of the context menu, the entire WiFi module is in deep sleep mode with all radio circuitry turned off.

Speaking of tin foils: If your C64 case still has the foil that covers the board, it might be necessary to remove it for a better WiFi signal.

MULTI MODE DISK DRIVE

6.1 Introduction

Since version 3.10 of the firmware, the Ultimate-II+ and Ultimate 64 have full hardware emulation support for 1541, 1571 *and* 1581. Previously, only the 1541 was implemented in the FPGA. Now, this drive has been replaced by a so called 'multi mode drive'. This multi-mode drive is an implementation that shares properties of the 1541, 1571 and 1581, which can be switched on or off on the fly. In practice this means that the hardware-emulated disk drive can be set to act like a 1541, a 1571 or 1581. This switching is fully managed by the application that runs on your Ultimate product. This chapter explains how to use it.

6.2 Manual Drive Type Selection

In the configuration menu (F2) you will find two drives, indicated by *Drive A Settings* and *Drive B Settings* respectively. Each drive can be of any of the aforementioned types; 1541, 1571 or 1581. To change the drive type, simply enter the configuration item by pressing CRSR-Right or RETURN. The second line of the configuration shows the selection:

Press enter and select the preferred drive type from the drop down list, or cycle through the options with the + and - keys. (Note that CRSR-Right does the same as +.) Once the configuration screen is closed – either by pressing RUN-STOP to leave the configuration screen entirely, or the CRSR-Left key combination to go up one level –, the drive mode is switched. The appropriate ROM is read into the drive, the sound samples are loaded, and the drive is reset. When the settings are saved in flash memory, this drive type is selected on boot.

6.3 Automatic Drive Type Selection

When browsing through your disk images, you may want to mount a .d81 while the current drive type is 1541. The .d81 format describes a 800kB 3.5" disk, intended to store a disk image of a 1581 disk. As you may know, a 3.5" disk does not fit into the 1541, so the drive needs to be changed into a 1581. Fortunately, it is not necessary to go to the configuration screen first. When trying to mount this disk, the application asks you if you like to change the drive type to 1581. When 'Yes' is chosen, the drive type is changed; the new ROM and appropriate sound samples are loaded and the drive is reset. Then the disk image is mounted.

Note that this is a temporary change. There is *no* change to the configuration settings.

6.4 Selecting Drive ROMs

When installing the 3.10 firmware, one ROM for each of the drives is installed as well. These binary files of either exactly 16K or exactly 32K in size are saved to the internal flash drive, in **/flash/roms**, named '1541.rom', '1571.rom' and '1581.rom' respectively. Installing other drive ROMs, like SpeedDos, DolphinDos or JiffyDos can be done in two ways:

1) Browse to the .bin or .rom file on your USB stick, and open the context menu by pressing RETURN. Then select 'Set as 15x1 ROM'. This will copy the ROM file to **/flash/roms** and automatically set the filename in the configuration of Drive A; the drive likely to be used most often.

2) Copy the .bin or .rom file on your USB stick manually to **/flash/roms**, by selecting it (press SPACE), then CTRL-C, navigate to **/flash/roms**, and then paste it there using CTRL-V. Then, go to the configuration menu (F2), go to Drive A or Drive B Settings, and use the drop-down context menu on the line 'ROM for . . .'. It will show you the files that are currently in the **/flash/roms** directory.
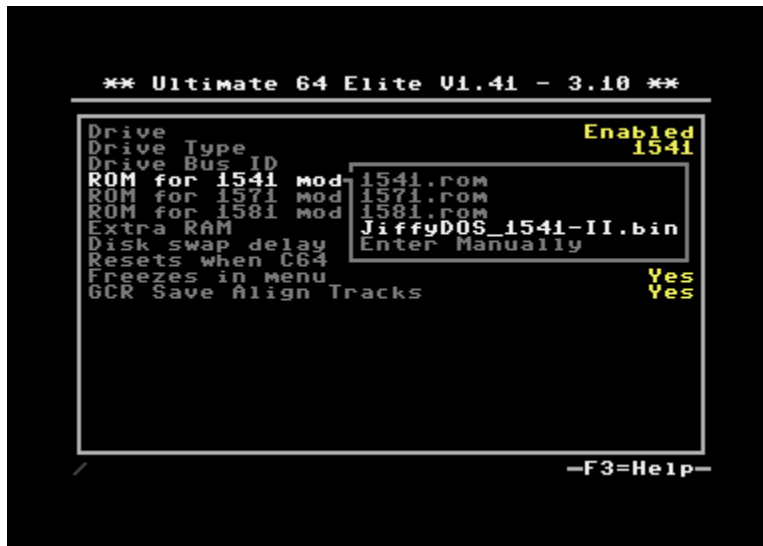
6.5 Other Settings

6.5.1 Drive Bus ID

With this option, the hardware bus ID can be set. Only the options 8, 9, 10 and 11 are available, because those are the ones that can be set in the hardware of these drives.

6.5.2 Extra RAM

Some ROMs require the drive to have more RAM. Standard, the 1541 and 1571 come with 2K SRAM memory; and the 1581 comes with 8K of RAM. This is not enough to store an entire raw track, which is required for some of the ROMs to speed up disk read/write performance. By enabling the 'Extra RAM' option, all locations that are not occupied by I/O devices and ROM will be available as RAM. Note that the I/O mirroring that the 1541 has will be eliminated as well; VIA chips will then *only* be visible from $1800-$1FFF, and no longer on $3800-$3FFF, $5800-$5FFF and $7800-$7FFF. Note that $8000-$FFFF is *always* ROM, in any of the drive types.

6.5.3 Disk Swap Delay

The 1541 and 1571 use the write protect light sensor to know whether the disk has been changed. For proper detection, these light pulses should be of suffient length. By default 100ms is enough, but you can change it here.

6.5.4 Resets when C64 resets

A reset signal is passed through the IEC (serial) cable. On some machines, this reset signal is triggered only when the C64 is turned on, while on others it triggers whenever the C64 is reset, e.g. by a button on the userport or expansion port. With this option, it is possible to reset the internal drives when a reset of the C64 occurs, regardless of the RESET# signal on the serial cable. When this option is turned off; only the RESET# signal on the serial cable is observed.

6.5.5 Freezes in menu

The C64 as well as the floppy drives have their own CPU. The Ultimate application enforces a full stop of the C64 CPU while displaying the menu on the VIC. (On the U64 this is only the case when the 'Freeze' mode is selected in the 'User Interface Settings' configuration menu.) More often than not, the C64 CPU and the drive CPU are synchronized to a certain extent. If the drive CPU would continue to run, while the C64 CPU is stopped, this could mean that the game or demo crashes. With this option, which is enabled by default, the drive CPU *also* stops when the C64 CPU is halted.

6.5.6 GCR Save Align Tracks

The 1541 and 1571 use Group Code Recording, or GCR to store data. This code provides enough flux transitions to reliabily detect the data, even despite small speed variations. This is the raw data format that the drive uses internally. Disk images can be stored in this format as well, called .g64 or .g71. This allows the representation of disk data that does not fully follow the default formatting rules.

Because commodore does not use the 'index pulse' a track can start at any angle on the disk. This would mean that when saving a track into a .g64 (or .g71) disk image, the data may start somewhere in the middle of a sector, while the beginning of that sector is stored at the end of the data track. Although perfectly valid, this makes decoding of a .g64 (or .g71) disk somewhat more complex. For this reason, the Ultimate can align these tracks when writing them into the disk image, which causes each track to properly start between sectors.

This option has no effect on the 1581, nor on .d64 and .d71 images.

CARTRIDGE EMULATION

Since firmware version 3.10, the way cartidges are emulated in the Ultimate has changed. While previously there were built-in cartridges, all cart roms are now optional. These previously built-in cartridges were stored in a flash chip on the board. These ROMs were taking up a lot of space; especially because nobody used all built-in cartridges much of that space was wasted. These built-in ROMs have been removed and the freed space was reassigned to the *flash drive*. The *flash drive* is now a permanent entry in the root of the file system and therefore always accessible. As a consequence, you can now use this space for your own cartridge of your liking, or one that is patched for your own needs. This chapter describes how to run and install your own cartridge roms, and how to switch between them.

7.1 Cartridge File Format

The file format that is used by the Ultimate is the standardized '.crt' format. This file format is defined and maintained by the VICE team. ".crt"-files do not only contain the raw ROM data, but also some meta data which is required to know how the ROM data needs to be interpreted. One of the important fields in this meta data is the "Hardware Type" field, which describes what type of cartridge it is. There is a long list of hardware types. Note that not all of them have been implemented in the Ultimate hardware.

7.2 Running a Cartridge ROM

At times, you may want to simply run a cartridge ROM temporarily, e.g. when you want to play a game. In this case, the cartridge can be started by simply navigating to the .CRT file and select "Run Cartridge" from the context menu. The cartridge will be loaded and 'attached' to the C64, and the machine is reset to start the cartridge. Resetting the machine will restart the cartridge again, just like what would happen with a real cartridge. To disable the cartridge, press the Task Menu key (F5), and select "C64 Machine" and then "Reboot". A reboot will install the permanently selected cartridge, if any, and reset the machine.

7.3 Installing a Cartridge ROM

For utility cartridges that need to start when the machine is turned on, the CRT file should be loaded at boot. This can be done by placing the CRT file onto the *flash drive*, in the **/flash/carts** directory. Note that this directory is hard coded and cannot be changed. There are two ways to get your CRT file in that directory:

1) Browse to the .crt file on your USB stick, and open the context menu by pressing RETURN. Then select 'Copy to Flash'. This will copy the ROM file to **/flash/carts** *and* automatically set the filename in the configuration. When the filename is too long, it will be truncated (prior to copying) to fit in the configuration field.

2) Copy the .crt file on your USB stick manually to **/flash/carts**, by selecting it (press SPACE), then CTRL-C, navigate to **/flash/carts**, and then paste it there using CTRL-V. Then, go to the configuration menu (F2), go to *C64 and Cartridge Settings*, and use the drop-down context menu on the line 'Cartridge'. It will show you the files with the .crt extension that are currently in the **/flash/carts** directory.
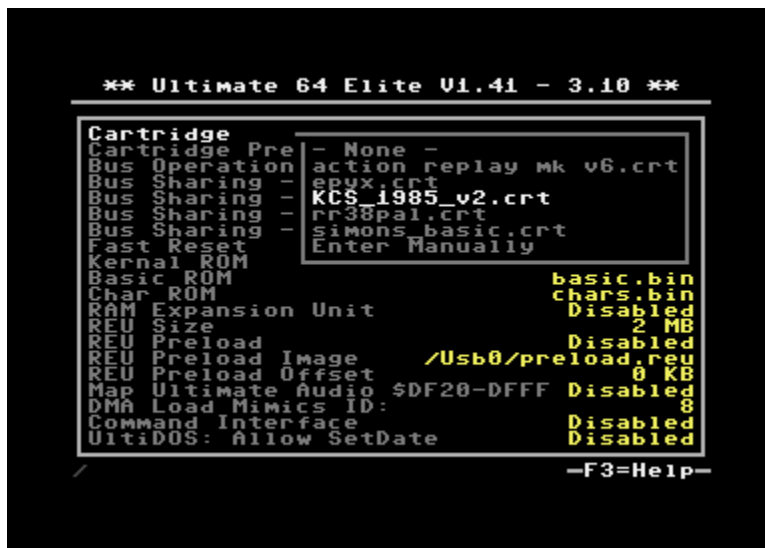
Note that the size of the internal flash is very limited. On the 1541U2, the usable space is about 700 KiB, on the U2+ the size is roughly 1500 KiB, and on the U64 the usable size of the internal flash is about 3500 KiB. This may limit the use of the flash to utility cartridges, roms and some small configuration data, and maybe a disk image that you may use often.

7.4 Supported C64 Cartridge Types

Type#	Description	1541U2	U2+	U64
0	Normal Cartridge (8K, 16K, UMAX)			
1	Action Replay			
2	KCS Power Cartridge			
3	Final Cartridge III			
4	Simons Basic			
5	Ocean type 1			
6	Expert Cartridge			
7	Fun Play			
8	Super Games			
9	Atomic Power			
10	Epyx Fastload			
11	Westermann			
12	Rex			
13	Final Cartridge I			
14	Magic Formel			
15	C64 Game System			
16	Warpspeed			
17	Dinamic			
18	Zaxxon			
19	Magic Desk, Domark, HES Australia			
20	Super Snapshot 5			
21	COMAL 80			
32	EasyFlash			
33	EasyFlash X-Bank			
34	Capture			
35	Action Replay 3			
36	Retro Replay			
37	MMC64			
38	MMC Replay			
39	IDE64			
40	Super Snapshot V4			
41	IEEE 488			
42	Game Killer			
43	Prophet 64			
44	EXOS			
45	Freeze Frame			
46	Freeze Machine			
47	Snapshot64			
48	Super Explode V5			
49	Magic Voice			
50	Action Replay 2			
51	MACH 5			
52	Diashow Maker			
53	Pagefox			
54	Kingsoft Business Basic			
55	Silver Rock 128			
56	Formel 64			

Table 1 – continued from previous page

Type#	Description	1541U2	U2+	U64
57	RGCD			
58	RR-Net MK3			
59	Easy Calc			
60	GMod2	†		
61	MAX Basic			
62	GMod3			
63	ZIPP-CODE 48			
64	Blackbox V8			
65	Blackbox V3			
66	Blackbox V4			
67	REX RAM-Floppy			
68	BIS-Plus			
69	SD-BOX			
70	MultiMAX			
71	Blackbox V9			
72	Lt. Kernal Host Adaptor			
73	RAMLink			
74	H.E.R.O.			

7.5 Supported C128 Cartridge Types

Type#	Description	1541U2	U2+	U64
0	C128 Cartridge $8000-$FFFF			
1	C128 Cartridge with I/O mirror			

7.6 Hardware Sub-Types

In one of the latest refinements of the .CRT file format, a field has been defined that specifies the "sub type" of a cartridge. The Ultimate uses this field in some cases to know whether the ROM has been made compatible with other I/O functions, such as the RAM Expansion Unit. The following table shows these cases. For other cartridge types, this field has no effect. Note, this field is one byte at CRT header offset 0x1A.

Type#	Sub	Description
C64:3	0	Standard FC-III cart (not compatible with REU)
C64:3	1	Patched FC-III ROM, free $DF00-$DF1F area for REU
C64:32	0	Standard EasyFlash Cart ROM (uses ROM at $DF00-DF1F)
C64:32	1	REU-aware EasyFlash Cart ROM ($DF00-$DF1F not used)
C128:1	0	Standard C128 cartridge with I/O mirror ($DE00-$DFFF)
C128:1	1	REU-aware C128 cartridge; with UCI preferred

7.7 EasyFlash

EasyFlash cartridges have a Flash chip that can be written from the C64 side. Games that are released on EasyFlash cartridge use the Flash to save settings and high scores to. The EasyFlash system is conceptually very advanced; as it supplies the 'driver' software to write to the ROM as part of the ROM data. This driver is called "EasyFlash API", or short 'EAPI'. The Ultimate supports these writes to the ROM by patching the EAPI code area with its own driver. This way, all titles that use the EAPI to write data to the flash will use the Ultimate code to do so. This way, the Flash chip itself does not need to be emulated. More background information about the EAPI can be found here: EasyFlash API.

It is required to manually save the EasyFlash title back to your storage device after it has been modified. To do so, press F5 inside of a writable directory and select C64 Machine –> Save Cartridge. This command will create a new CRT file with all ROM data.
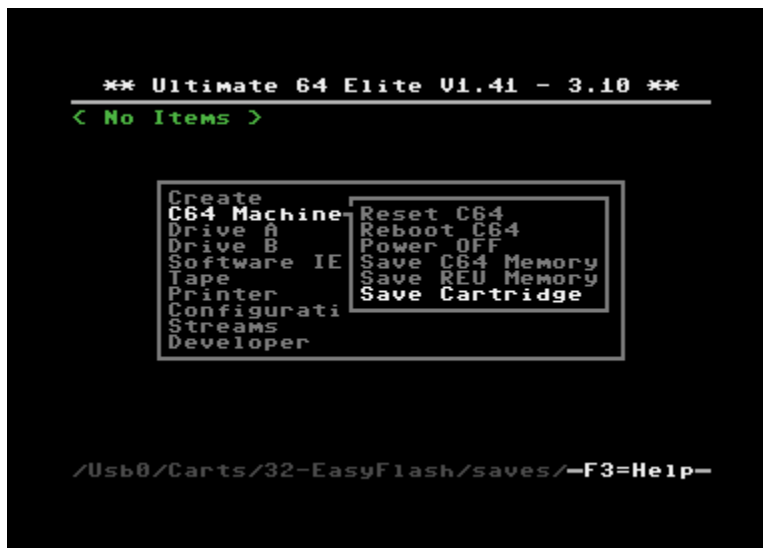
7.8 GMod2

Also GMod2 cartridges have a chip to store data in; an M93C86 EEPROM chip. This chip is fully emulated in the Ultimate*[0]. The GMod2 software can 'see' this EEPROM chip, and use it in the standard way. If the contents are part of the CRT file, modified contents of this EEPROM can also be saved. To do this, follow the same procedure as with EasyFlash. *Note that when the EEPROM data is _not_ part of the original CRT, it will be initialized with 'FF' bytes, and it will _not_ be saved with the Save Cartridge method, even when new data was written to it.*

7.9 Compatibility with I/O features

Unfortunately the Commodore 64 was never fully 'plug and play', as there was no configuraiton managment and dynamic allocation of memory and I/O resources like on modern PCs. This means that cartridges may be incompatible with some software, but also that enabling different features on the cartridge port may cause conflicts. In version 3.10 of the firmware, an attempt has been made to automatically 'fix' compatibility issues, by disabling features that conflict with the selected cartridge.

In order to find out what features were automatically disabled (or even enabled in a few cases), you can open up the System Info page with F4. An example is shown here:

---

[0] There is one special version of the 1541U2 that supports GMod2, at the cost of other features.

As can be seen from the image, the EasyFlash cartridge that was loaded is not compatible with the RAM Expansion Unit (REU), nor with the Ultimate Command Interface (UCI) and with the Ultimate Audio Sampler module. All these reside in the $DF00-$DFFF area, which is also used by the EasyFlash cartridge. This means that the priority has been given to a correct functioning of the EF cartridge.

EIGHT

TAPE SUPPORT

8.1 TAP

The Ultimate products are capable of emulating a cassette tape deck (CBM1530/1531). To use this function on the Ultimate-II+, a special adapter set connects the cartridge to the tape port of your C=64 computer. This tape adaptor set can be purchased separately. On the Ultimate 64, such a cable is not required, as it is built-in.

In order to play a tape, browse in the menu to a .TAP file, and press enter and select "Play Tape" from the popup menu. This will initialize the tape streamer from the start of the tape. Use functions in the main menu (F5) to pause/resume playback. It is also possible to write the .TAP file to a real tape, using a CBM1530/1531 deck.

Tape signals can also be captured into a .TAP file. The 'F5' menu will show you the available options.

8.2 T64

How do I use T64 files?

T64 is a file format, or rather a container, just like D64 disk image. It can contain multiple PRG files. You can simply enter the container by pressing CRSR-RIGHT to see the contents and start the PRG files using the contextual menu pop up, by choosing to run the c64 program.

Applies to: Ultimate 1541-II, Ultimate 1541-II+, Ultimate 64

MODEM SUPPORT

9.1 Introduction

Starting from version 3.7, a lightweight modem emulation layer is provided in the firmware of the Ultimate. This modem layer is accessible through an emulated MOS 6551 ACIA chip. This chip was found in the SwiftLink cartridge, as well as some other ACIA based RS-232 cartridges of the time. This ACIA chip provides a virtual serial port to the C-64, which is accessed through the I/O space.

The modem emulation layer bridges the ACIA chip and the LAN port. The modem connects to a server (e.g. one that is running a bulletin board system) through the internet.

9.2 Configuration

In order to use the modem emulation layer, the ACIA needs to be enabled in the configuration menu of the Ultimate application. Secondly, a the terminal program needs to be used to access the serial port, e.g. CCGMS.

An example configuration could be:
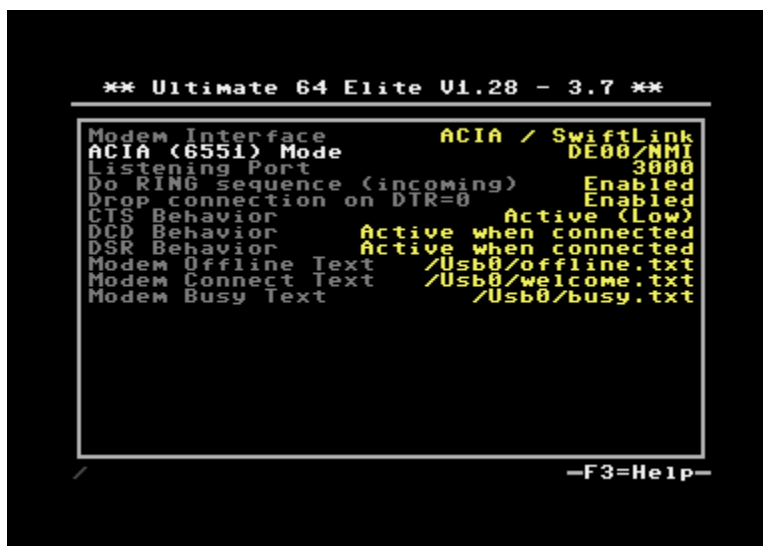
Please observe I/O range conflicts, this is *not* enforced by the Ultimate application.

Once the ACIA chip is enabled in the I/O space, a terminal program can be started and configured. For example, the configuration of CCGMS will be shown here:

Note that CCGMS uses the NMI line, and the SwiftLink is configured at $DE00. This corresponds to the setting made in the Ultimate configuration menu.

Again: Please observe I/O range conflicts, this is *not* enforced by the Ultimate application. If you are getting wrong characters in your terminal program, you likely have an I/O conflict. For instance, Retro Replay has registers at $DE00, and will therefore conflict with the ACIA at $DE00.

9.3 Modem Commands

The lightweight modem emulation supports a subset of the standard Hayes modem commands. These commands start with 'AT'.

Command	Description
ATI	Identify. This command prints the modem identification text message.
ATZ	Reset. This command resets the modem. Any existing connection will be dropped.
ATH	Hangup. This command terminates the current connection.
ATD	Dial. With this command an outgoing connection is initiated. The 'D' should be followed by another character, usually 'T' or 'P' for tone and pulse dialing. However, the Ultimate ignores this character. The domain name follows. The port number can be specified after a colon. This is optional; when the port number is not given, the Ultimate will attempt to connect to port 80. An example of such command is: *ATDTAFTER-LIFE.DYNU.COM:6400*
ATA	Answer. This command picks up the incoming call. This is a required command when the option 'Do RING sequence' is set to 'Enabled'. If this command is not given in time, the incoming call times out after a number of rings.
ATO	Online. Use this command to go back to an active connection, if it was interrupted by the +++ sequence.
ATV	Verbose mode. Recognized but ignored. Usually, such a command appears in the initialization string of a terminal program, such as StrikeTerm. It is followed by a digit.
ATS	Register Select. With this command the so called 'S'-registers can be read and set. Not all registers are supported, but some useful registers are S0 (auto answer), S1 (ring counter), S2 (escape char) and S12 (escape time). See Hayes modem specification for more details.
+++	Escape sequence. The actual character can be set with register S2, with defaults to '+'. When three of these characters are sent to the modem, and at least S12 'jiffies' expire (by default 50, thus one second), the modem switches to command mode, but the existing connection remains active. A command such as ATH can then be given.

9.4 Incoming connections

Incoming connections are also supported. This enables you to run a simple server on your C64, or maybe even a BBS! In the configuration can be seen that port 3000 is selected for incoming connections. When an external node on the network attempts to connect to TCP port 3000 of the Ultimate, this is recognized as an incoming connection. Depending on the current state of the modem, the following will happen:

State	Behavior
Offline	When DTR=0, it is assumed that the modem software is not running. This happens when the ACIA is not configured and thus not enabled. In this case, the connecting party will receive a message, which is defined by the file specified in the configuration (default: */Usb0/offline.txt*). If this file does not exist, the default message **Modem Software is currently not running…** is answered.
Busy	When the modem is currently already in a call, thus the modem has an active connection, the connecting party will receive a message, which is defined by the file specified in the configuration (default: */Usb0/busy.txt*). If this file does not exist, the default message **The modem you are connecting to is currently busy.** is answered.
Ready	The modem is configured and the software is ready to accept a call. In this case, the connecting party will receive a message, which is defined by the file specified in the configuration (default: */Usb0/connect.txt*). If this file does not exist, the default message **Welcome to the Modem Emulation Layer of the Ultimate!** is answered. Following this message the RING sequence will begin (if enabled in the configuration). This means that the terminal program will receive 'RING' messages, which it needs to answer with 'ATA' to answer.

Note, that when the filename is set to an empty string, the message is suppressed; thus no message is being answered to the calling party.

9.5 Handshaking

In the configuration, the state of handshake lines that the terminal program 'sees' can be set. The state of these handshake lines are set when a connection is established or terminated. At these moments, the Ultimate will read the configuration and set the handshake lines accordingly. By default CTS is always active, and DCD and DSR are only active during an active connection.

COMMAND INTERFACE

10.1 Introduction

10.1.1 Context

The 'Ultimate-II command interface' feature is a feature for the 1541 Ultimate-II module. It implements a way to communicate to the 1541 Ultimate-II programmatically from the Commodore 64, through the cartridge port I/O registers. This 'Ultimate-II command Interface' is also abveviated as 'UCI'. It is also implemented on the Ultimate-II+ and Ultimate-64 products.

The 'Ultimate-II command interface' is the communication layer between the C64 on one side and functional modules of the 1541 Ultimate-II / Ultimate-II+ / Ultimate 64 on the other side. Such a functional module takes a command, and returns data and status back to the user program on the C64.

There are a few functional modules, which are described in their own separate documents:

- "Ultimate DOS"; meant to open, read and write files and directories

- "Control Interface"; a collection of useful routines and calls to access U2/U2+/U64 features

- "Network Target"; an interface to directly access the TCP/IP network stack.

10.1.2 Purpose of this document

The 'Ultimate-II command interface' feature is accessible from the cartridge I/O range. In this manual, the programming API is described.

10.2 Transport Layer

10.2.1 Overview

The transport layer of the Ultimate-II Command Interface makes use of a register interface which is accessible through the cartridge port of the C64. The registers are mapped into I/O space, at the address $DF1C up to $DF1F, masking the last four registers of the RAM Expansion Unit (REU). Mapping this small 4-byte block into the I/O space is optional, and needs to be turned on in the 'Command Interface' configuration menu.

10.2.2 Register Overview

The following table shows the four registers and their meaning.

Address	Description	Default
$DF1C	Control register (Write)	
$DF1C	Status register (Read)	$00
$DF1D	Command data register (Write)	
$DF1D	Identification register (Read)	$C9
$DF1E	Response Data register (Read only)	
$DF1F	Status Data register (Read only)	

10.2.3 Communication Protocol Basics

The communication protocol is based on a state machine with four states:

- Idle

- Command Busy

- Data Last (last block)

- Data More (more data pending)

The transition through these states takes place in an interaction between the C64 user software and the software running on the Ultimate-II. The state that the communication protocol is in can be read from the Status register at $DF1C.

Transfer command

When the protocol is in idle state (see paragraph 2.4.2 for the state encoding), the Ultimate-II is ready to receive a new command. This is done by writing the command byte by byte into the command data register at $DF1D. Then, the command is pushed into the Ultimate-II by writing a '1' to the control bit 'PUSH_CMD', in the control register. This will cause a state transition to "Command Busy".

Ultimate-II processes command

As soon as the Ultimate-II has decided how to respond to this command, it will prepare the data and status reply, and cause the state machine to move to one of the two data states: 'Data Last', or 'Data More'.

Reading the reply

The user software can now read both data and status from the respective registers $DF1E and $DF1F. Whether there is data or status available can be seen from the upper two bits of the status register. These bits will be '1' when there is still more data to be read, and '0' otherwise.

Release: Accept data

As soon as the software has read all the data (or decides not to do so), the C64 should write a '1' to the register bit 'DATA_ACC', to indicate that all data was accepted. If this was the last data block, this causes the state machine to go back to the idle state, or else, the state returns to "Command Busy".

State machine

The following diagram shows the state machine:
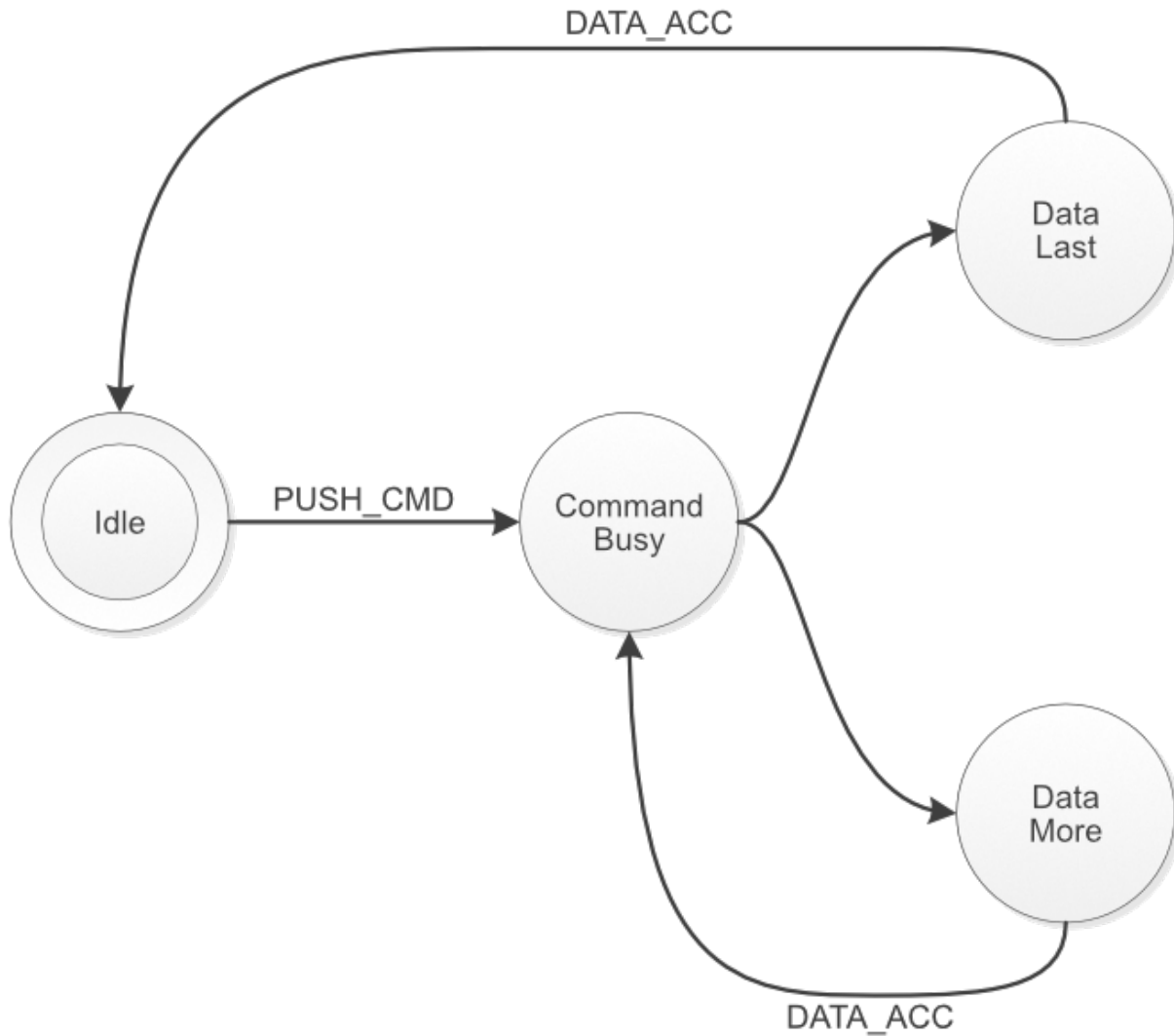
Figure 1: Basic Protocol State Machine

Note: Setting the 'ABORT' bit, does not directly influence the state machine. It is handled by the Ultimate-II software, which will in turn reset the state machine to idle eventually.

10.3 Register Details

10.3.1 Control Register

The control register contains the following bits:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
reserved				CLR_ERR	ABORT	DATA_ACC	PUSH_CMD

PUSH_CMD: Writing a '1' to this register bit causes the command that was written to the command data register to be pushed into the software of the Ultimate-II.

DATA_ACC: Writing a '1' to this register bit tells the communication layer that all data from the Ultimate-II was accepted. This is automatically ignored when the communication controller is not in one of the two data states. Writing to this bit also causes the transfer of the data/status queues to be aborted and reset. Thus, the data response and status response queues will be empty after writing this bit.

ABORT: Writing a '1' to this register sets the 'abort' flag in the communication controller. This bit is polled by the Ultimate-II software. When it finds this bit set, the current communication is aborted, and the state machine is forced back to the idle state.

CLR_ERR: Pushing a command to the Ultimate-II when the communication layer is not in idle mode, causes a state error flag to be set. See status register. Write a '1' to CLR_ERR to clear this error condition.

Status register

The status register contains the following bits:

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DATA_AV	STAT_AV	*STATE*		ERROR	ABORT_P	DATA_ACC	CMD_BUSY

CMD_BUSY: This bit indicates that there is a pending command in the command memory.

DATA_ACC: This bit reflects the condition that the user has told the Ultimate-II that it accepted the data.

ABORT_P: This bit reflects the state of the internal abort flag. When this bit is '1', the Ultimate-II still has to handle the abort request.

ERROR: When this bit is '1', the user tried to send a command to the Ultimate-II while it was not in idle state.

STATE: These two bits encode the protocol state:

00: Idle

01: Command Busy

10: Data Last

11: Data More

STAT_AV: When this bit is '1', there is status data available from the status queue, accessible through the status data register ($DF1F).

DATA_AV: When this bit is '1', there is response data available from the data queue, accessible through the response data register ($DF1E).

10.3.2 Queues

As previously described, there are three byte-queues that the Ultimate-II Command Interface uses:

- Command queue
- Response Data queue
- Status queue

The sizes of these queues are important to note, since they define the maximum transfer size per command. The command queue size is 896 bytes ($380), the response data queue is also 896 bytes ($380), and the status queue is 256 bytes ($100).

10.4 Dispatch Layer

On top of the transport layer, there is light weight dispatcher. This dispatcher sends the command from the user software to a functional module in the 1541 Ultimate-II. The first byte of the command is determines the destination. Such a destination is called a 'target'.

Initially, in version 2.6 of the Ultimate-II firmware, there is only one functional target: "Ultimate-DOS". Two instances of this DOS are located at targets 1 and 2. See the documentation of this target to obtain more information on the commands this target implements.

ULTIMATE-II VIRTUAL PRINTER

By René Garcia Version: February 18th 2018 All rights reserved.

11.1 Introduction

11.1.1 Context

The virtual printer is an Ultimate-II feature since 3.0 firmware. With this functionality you can print from your Commodore 64/128 using a virtual IEC device #4 or #5.

This emulation simulates a Commodore MPS-1230 printer with all the commands that this printer can understand. Not all commands are executed as some of them are hardware related and cannot obviously be implemented. The results are printed to PNG image files, one file per page. You can also choose to bypass the printer emulation and to send the raw data from #4 or #5 IEC device to a file.

MPS-1230 was a mid-range black ink ribbon 9 needle matrix printer sold by Commodore in the late 80's.

This printer is compatible with nearly all the usual programs that have been edited for C64/C128. It can interpret 4 printer instruction sets:

- Commodore MPS-801
- Epson FX-80
- IBM Graphics Printer
- IBM Proprinter

11.1.2 License

Virtual Printer is released under the GNU General Public License 3.0. A full copy of the license is included in the root of the Ultimate-II firmware sources.

11.1.3 Purpose of this document

This document describes how to use and configure the Ultimate-II embedded virtual printer.

You will also find all the commands and charsets supported by the printer. Then you can add printer facility to your own BASIC programs!

11.2 Configuration

11.2.1 Overview

You will find all the configuration items for the printer in the IEC configuration menu.

11.2.2 Enable the printer

To enable the printer, you need to enable the software IEC feature in the Ultimate-II:

- Use the F2 Menu to enter Ultimate-II configuration and then select "**Software IEC Settings**"

- Then on item "**IEC Drive and Printer**" select "**Enabled**"

11.2.3 Printer configuration items
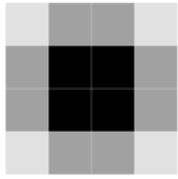
- **Printer Bus ID**: 4 or 5 (default is 4)

  This will assign device ID 4 or 5 to the printer.

- **Printer output file**: (default is */SD/*printer on Ultimate II or */Usb0/printer* on Ultimate II+)

  You can select file base name that the virtual printer will use to create the PNG files. If you choose to generate PNG files they will be named *printer-001.png*, *printer-002.png*, and so on. If you chose the bypass the emulation and write RAW binary data to disk the file will be named *printer* with no extension. When using ASCII filter output, extension *.txt* will be appended to file name.

- **Printer output type**: PNG, ASCII or RAW (default is PNG)

  PNG are images created by the printer emulator each time a page is ejected from the printer. Caution, if a file with the same name already exists, it will not be overwritten and the page is lost. RAW is the data directly sent by the C64/128 to the IEC port and recorded as binary to a file. ASCII will keep and convert printable characters to ISO8859-1 standard. This output only makes sense if you are printing text as you will only get garbage with bitmap. In both RAW and ASCII output mode, if the file already exists, the new data will be appended to it.

- **Printer ink density**: Low, Medium or High (default is Medium)

  You can consider this as "how strong is the pin impact on the paper". *Low* will only print very small dots and *High* larger dots. As a consequence, this will change the resulting contrast. *High* gives the best result for DRAFT character mode. *Medium* may be well suited for NLQ character mode. Just test and see what match your needs. *See table below for samples*.

- **Printer emulation**: Commodore MPS, Epson FX-80, IBM Graphics Printer, IBM Proprinter (default is Commodore MPS)

  You can select which instruction set the emulator will recognize. Changing from one emulation to another will reset the printer attributes but the printer head stays at the same place and the page is not ejected.

- **Printer Commodore charset:** USA/UK, Denmark, France/Italy, Germany, Spain, Sweden, Switzerland (default is USA/UK)

Select which charset to use when using Commodore MPS emulation. If you don't know which one to choose, USA/UK is the one you want. See Commodore charset description on chapter 19.

- **Printer Epson charset:** Basic, USA, France, Germany, England, Denmark I, Sweden, Italy, Spain, Japan, Norway, Denmark II (default is Basic)

  Select which charset to use when using Epson FX-80 emulation. See Epson charset description on chapter 10.

- **Printer IBM table 2:** International 1, International 2, Israel, Greece, Portugal, Spain (default is International 1)

  Select which charset to use for Table2 when using IBM Graphics Printer or IBM Proprinter emulation. IBM printers can use 2 charsets: Table 1 and Table2. Table 1 cannot be modified and is the default charset. Table 2 is the one you chose with this parameter. See IBM charset description in chapter 11.

Ink Density	Low	Medium	High
Elementary Dot (x1)	.	.	.
Elementary Dot (x300)			
Draft text	1541 ULTIMATE II	1541 ULTIMATE II	1541 ULTIMATE II
NLQ text	1541 ULTIMATE II	1541 ULTIMATE II	**1541 ULTIMATE II**
Draft graphic chars	♠ ♥ ♦ ♣	♠ ♥ ♦ ♣	♠ ♥ ♦ ♣
NLQ graphic chars	♣ ♥ ♦ ♣	♠ ♥ ♦ ♣	♠ ♥ ♦ ♣

11.3 Using the printer

11.3.1 Printing from the C64/C128

Just use your program and tell it that you have a connected printer compatible with MPS Commodore series (e.g.: MPS-801/MPS-803 are the most frequently supported commodore printers).

11.3.2 Flushing the printer spool

The printer has a very small buffer (256 bytes) and some data may still be in the buffer waiting to be printed when your print job is finished. The printer doesn't know that your job is finished and waits for more data to print until the end of the page.

You need to tell the printer that you want all the buffered data to be printed and to eject the current page. This works as the *Form Feed* button on the real MPS-1230 to eject the page.

Go to F5 Menu and select "**Flush Printer/Eject Page**". In PNG mode, this will make the current page to be written to a file. Next print job will start on a blank page. In RAW and ASCII mode this will write the buffered data to the file.

11.3.3 Resetting the printer

You may need to reset printer to go back to an initial state. Go to F5 Menu and select "**Reset IEC and Printer**". Current data in printer buffer is lost. Current page that was being printed is also lost.

11.3.4 Performances

Composing a page full of text and creating the PNG file will need approximatively 15 seconds on the Ultimate-II (28 seconds using NLQ mode). You may think it's slow but this is much faster than a real MPS-1230 printer (1 min in DRAFT mode, 4 min in NLQ mode) !

The Ultimate-II middle button becomes unresponsive while composing a page. The green LED on the right of the cartridge is lit when printer is working. Be patient and look at the activity LED to stop blinking.

RAW and ASCII modes are nearly immediate. There is no process time to wait.

At this time, with firmware 3.2, The virtual printer is slower on Ultimate II+ than on Ultimate II as it is using a slower CPU. In fact, no processor cache is implemented yet in Ultimate II+, this may change in a future firmware as CPU is implemented in FPGA using VHDL.

11.4 Capabilities

This table summarize the printer capabilities depending on which printer emulation is active:

	Commodore MPS	Epson FX-80	IBM Graphics Printer	IBM Proprinter
Draft	•	•	•	•
Double strike	•	•	•	•
Bold	•	•	•	•
Italic (draft only)	•	•	•	
NLQ	•	•	•	•
Underline	•	•	•	•
Double width	•	•	•	•
Superscript	•	•	•	•

Table 1 – continued from previous page

	Commodore MPS	Epson FX-80	IBM Graphics Printer	IBM Proprinter
Subscript	•	•	•	•
Reverse	•			
Overline				•
Backspace		•	•	•
Reverse page feed		•		
CR=CR+LF	•			*optional*
LF=CR+LF	•	•		
7 dot BIM	•			
8 dot BIM		•	•	•
9 dot BIM		•		
HT Program		•	•	•
VT Program		•		•
60 dpi BIM	• *(double width)*	•	•	•
75 dpi BIM		•		
80 dpi BIM		•		
90 dpi BIM		•		

Table 1 – continued from previous page

	Commodore MPS	Epson FX-80	IBM Graphics Printer	IBM Proprinter
120 dpi BIM		•	•	•
240 dpi BIM		•	•	•
Pica (10cpi)	•	•	•	•
Elite (12cpi)	•	•	•	•
Micro (15cpi)	•			
Condensed (17.1cpi)	•	•	•	•
Pica Compressed (20cpi)	•			
Elite Compressed (24 cpi)	•			
Micro Compressed (30 cpi)	•			

11.5 Commodore MPS commands

This chapter describes the commands the printer can understand when using the Commodore MPS emulation. You will find Commodore BASIC examples to explain you how to use them. This printer uses PETASCII.

11.5.1 Simple example

This will print a first line with HELLO WORLD! on it and a second line with HELLO printed with double width characters.

10 OPEN1,4

20 PRINT#1,"HELLO WORLD!"

30 PRINT#1,CHR$(14)"HELLO"

40 CLOSE1

HELLO WORLD!
HELLO

11.5.2 Secondary address

Only on Commodore MPS emulation, you can specify an optional secondary address on OPEN :

- **0** : Select PETASCII charset with uppercases and graphic chars
- **7** : Select PETASCII charset with lowercases and uppercases

If no secondary address is specified, 0 is the default.

11.5.3 Commands

For a detailed description of all commands, please refer to the PDF version of this document.

11.6 PETASCII character table

11.6.1 USA/UK

Table : USA/UK Charset in Uppercase/Graphic Mode (Secondary address = 0)

```
  | 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
--+-----------------------------------------------
0 |       0  @  p  —  P              ┌  —  P        ┌
1 |    !  1  a  q  A  Q           █  ┴  A  Q  █     ┴
2 |    "  2  b  r  B  R           ■  ┬  B  R  ■     ┬
3 |    #  3  c  s  C  S           —  ┤  C  S  —     ┤
4 |    $  4  d  t  D  T           _  │  D  T  _     │
5 |    %  5  e  u  E  U           │  ▐  E  U  │     ▐
6 |    &  6  f  v  F  V           ▓  █  F  V  ▓     █
7 |    '  7  g  w  G  W           │  ─  G  W  │     ─
8 |    (  8  h  x  H  X           ▒  ▀  H  X  ▒     ▀
9 |    )  9  i  y  I  Y           ░  ▄  I  Y  ░     ▄
a |    *  :  j  z  J  Z           │  ✓  J  Z  │     ✓
b |    +  ;  k  [  K  +           ├  ▪  K  +  ├     ▪
c |    ,  <  l  £  L  ▓           ▪  ▫  L  ▓  ▪     ▫
d |    —  =  m  ]  M  │           └  ┘  M  │  └     ┘
e |    .  >  n  ↑  N  ⋮⋮          ┐  ▫  N  ⋮⋮ ┐     ▫
f |    /  ?  o  ←  O  ⟋⟋          ▬  ▪▪ O  ⟋⟋ ▬     ⋮⋮
```

Table USA/UK Charset in Lowercase/Uppercase Mode (Secondary address = 7)

11.6.2 Denmark

```
  | 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
--+-----------------------------------------------
0 |       0  @  P  —  ┐              ┌  —  ┐        ┌
1 |    !  1  A  Q  ♠  ●           █  ┴  ♠  ●  █     ┴
2 |    "  2  B  R  │  —           ■  ┬  │  —  ■     ┬
3 |    #  3  C  S  —  ♥           —  ┤  —  ♥  —     ┤
4 |    $  4  D  T  ─  │           _  │  ─  │  _     │
5 |    %  5  E  U  ─  ╱           │  ▐  ─  ╱  │     ▐
6 |    &  6  F  V  —  ✕           ▓  █  —  ✕  ▓     █
7 |    '  7  G  W  │  ○           │  ─  │  ○  │     ─
8 |    (  8  H  X  │  ♣           ▒  ▀  │  ♣  ▒     ▀
9 |    )  9  I  Y  ╲  │           ▰  ▄  ╲  │  ▰     ▄
A |    *  :  J  Z  ╲  ◆           │  ┘  ╲  ◆  │     ┘
B |    +  ;  K  Æ  ╱  +           ├  ▪  ╱  +  ├     ▪
C |    ,  <  L  Ø  └  ▓           ▪  ▫  └  ▓  ▪     ▫
D |    —  =  M  Å  ╲  │           └  ┘  ╲  │  └     ┘
E |    .  >  N  ↑  ╱  π           ┐  ▫  ╱  π  ┐     ▫
F |    /  ?  O  ←  ┌  ◣           ▬  ▪▪ ┌  ◣  ▬     π
```

Table : DENMARK Charset in Uppercase/Graphic Mode (Secondary address = 0)

```
  | 0 1 2 3 4 5 6 7 8 9 a b c d e f
--+--------------------------------
0 |       O @ p — P       ⌐ — P     ⌐
1 |     ! 1 a q A Q     ▌ ⊥ A Q ▌ ⊥
2 |     " 2 b r B R     ▬ ⊤ B R ▬ ⊤
3 |     # 3 c s C S     ⎯ ⊣ C S ⎯ ⊣
4 |     $ 4 d t D T     _ | D T _ |
5 |     % 5 e u E U     | ▌ E U | ▌
6 |     & 6 f v F V     ▓ ▌ F V ▓ ▌
7 |     ' 7 g w G W     | ⎯ G W | ⎯
8 |     ( 8 h x H X     ≈ ▀ H X ≈ ▀
9 |     ) 9 i y I Y     ⧄ ▄ I Y ⧄ ▄
a |     * : j z J Z     | ∨ J Z | ∨
b |     + ; k æ K Æ     ⊦ ▪ K Æ ⊦ ▪
c |     , < l ø L Ø     ▪ ▪ L Ø ▪ ▪
d |     - = m å M Å     ∟ ⌐ M Å ∟ ⌐
e |     . > n ↑ N ⌗     ⌐ ▪ N ⌗ ⌐ ▪
f |     / ? o ← O ≋     ▄ ▪ O ≋ ▄ ⌗
```

Table DENMARK Charset in Lowercase/Uppercase Mode (Secondary address = 7)

11.6.3 France / Italy

```
  | 0 1 2 3 4 5 6 7 8 9 A B C D E F
--+--------------------------------
0 |       O @ P ` ⌐       § ` ⌐     §
1 |     ! 1 A Q ◥ ◤     | à ◥ ◤ | à
2 |     " 2 B R ▬ ⌐     ∟ è ▬ ⌐ ∟ è
3 |     # 3 C S ▬ ▪     ⊥ ì ▬ ▪ ⊥ ì
4 |     $ 4 D T ▪ ⌐     ⌐ ò ▪ ⌐ ⌐ ò
5 |     % 5 E U ▪ ▌     ⊦ ù ▪ ▌ ⊦ ù
6 |     & 6 F V ` ▬     — â ` ▬ — â
7 |     ' 7 G W ⌐ ▪     ⊣ ê ⌐ ▪ ⊣ ê
8 |     ( 8 H X ▌ ▬     ⌐ î ▌ ▬ ⌐ î
9 |     ) 9 I Y | _     ⊤ ô | _ ⊤ ô
A |     * : J Z | ▌     ⌐ û | ▌ ⌐ û
B |     + ; K [ ⟍ ë     + Ä ⟍ ë + Ä
C |     , < L \ ⌐ ï     £ Ö ⌐ ï £ Ö
D |     - = M ] ⟋ °     ' Ü ⟋ ° ' Ü
E |     . > N ↑ ⎯ π     ^ ß ⎯ π ^ ß
F |     / ? O — ⌐ ç     ¨ é ⌐ ç ¨ π
```

Table : FRANCE/ITALY Charset in Uppercase/Graphic Mode (Secondary address = 0)

```
  | 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
--+------------------------------------------------
0 |          O  @  p  `  P           ß  `  P     ß
1 |       !  1  a  q  A  Q        |  à  A  Q  |  à
2 |       "  2  b  r  B  R        └  è  B  R  └  è
3 |       #  3  c  s  C  S        ┴  ì  C  S  ┴  ì
4 |       $  4  d  t  D  T        ┘  ò  D  T  ┘  ò
5 |       %  5  e  u  E  U        ├  ù  E  U  ├  ù
6 |       &  6  f  v  F  V        —  â  F  V  —  â
7 |       '  7  g  w  G  W        ┤  ê  G  W  ┤  ê
8 |       (  8  h  x  H  X        ┌  î  H  X  ┌  î
9 |       )  9  i  y  I  Y        ┬  ô  I  Y  ┬  ô
a |       *  :  j  z  J  Z        ┐  û  J  Z  ┐  û
b |       +  ;  k  [  K  ë        +  ä  K  ë  +  ä
c |       ,  <  l  \  L  ï        £  ö  L  ï  £  ö
d |       -  =  m  ]  M  °        '  ü  M  °  '  ü
e |       .  >  n  ↑  N  π        ^  ß  N  π  ^  ß
f |       /  ?  o  —  O  ç        ¨  é  O  ç  ¨  π
```

Table FRANCE/ITALY Charset in Lowercase/Uppercase Mode (Secondary address = 7)

11.6.4 Germany

```
  | 0  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
--+------------------------------------------------
0 |          O  @  P  `  ┐           @  `  ┐     `
1 |       !  1  A  Q  ◣  ◤        |  µ  ◣  ◤  |  ◣
2 |       "  2  B  R  ▬  ⁄        └  à  ▬  ⁄  └  ▬
3 |       #  3  C  S  ▬  ▪        ┴  ù  ▬  ▪  ┴  ▬
4 |       $  4  D  T  ▪  ╲        ┘  â  ▪  ╲  ┘  ▪
5 |       %  5  E  U  ▪  ▮        ├  ê  ▪  ▮  ├  ▪
6 |       &  6  F  V  ╲  ▬        —  î  ╲  ▬  —  ╲
7 |       '  7  G  W  ⁄  ▪        ┤  ô  ⁄  ▪  ┤  ⁄
8 |       (  8  H  X  ▮  ▬        ┌  û  ▮  ▬  ┌  ▮
9 |       )  9  I  Y  |  _        ┬  √  |  _  ┬  |
A |       *  :  J  Z  |  ▮        ┐  Σ  |  ▮  ┐  |
B |       +  ;  K  [  ╲  ▨        +  Ä  ╲  ▨  +  ╲
C |       ,  <  L  \  └  ┘        é  Ö  └  ┘  é  └
D |       -  =  M  ]  ⁄  ▓        £  Ü  ⁄  ▓  £  ⁄
E |       .  >  N  ↑  ▬  π        è  ß  ▬  π  è  ▬
F |       /  ?  O  —  ┌  _        '  ^  ┌  _  '  π
```

Table : GERMANY Charset in Uppercase/Graphic Mode (Secondary address = 0)

```
  | 0 1 2 3 4 5 6 7 8 9 a b c d e f
--+--------------------------------
0 |     0 @ p ` P       § ` P   §
1 |   ! 1 a q A Q     | à A Q | à
2 |   " 2 b r B R     ⌐ è B R ⌐ è
3 |   # 3 c s C S     ⊥ ì C S ⊥ ì
4 |   $ 4 d t D T     ⌐ ò D T ⌐ ò
5 |   % 5 e u E U     ⊢ ù E U ⊢ ù
6 |   & 6 f v F V     — â F V — â
7 |   ' 7 g w G W     ⊣ ê G W ⊣ ê
8 |   ( 8 h x H X     ┌ î H X ┌ î
9 |   ) 9 i y I Y     ┬ ô I Y ┬ ô
a |   * : j z J Z     ┐ û J Z ┐ û
b |   + ; k [ K Ä     + ä K Ä + ä
c |   , < l \ L Ö     é ö L Ö é ö
d |   - = m ] M Ü     £ ü M Ü £ ü
e |   . > n ↑ N π     è ß N π è ß
f |   / ? o ─ O _     ' é O _ ' π
```

Table GERMANY Charset in Lowercase/Uppercase Mode (Secondary address = 7)

11.6.5 Spain

```
  | 0 1 2 3 4 5 6 7 8 9 A B C D E F
--+--------------------------------
0 |     0 @ P ` ┐       ← ` ┐   ←
1 |   ! 1 A Q ◣ ◢     | À ◣ ◢ | À
2 |   " 2 B R ▬ ╱     ⌐ È ▬ ╱ ⌐ È
3 |   # 3 C S ▬ ▪     ⊥ ♠ ▬ ▪ ⊥ ♠
4 |   $ 4 D T ▪ ╲     ⌐ Ò ▪ ╲ ⌐ Ò
5 |   % 5 E U ▪ ▮     ⊢ ♥ ▪ ▮ ⊢ ♥
6 |   & 6 F V ╲ ▬     — Á ╲ ▬ — Á
7 |   ' 7 G W ╱ ▪     ⊣ É ╱ ▪ ⊣ É
8 |   ( 8 H X ▮ ▬     ┌ Í ▮ ▬ ┌ Í
9 |   ) 9 I Y | _     ┬ Ó | _ ┬ Ó
A |   * : J Z | ▮     ┐ Ú | ▮ ┐ Ú
B |   + ; K [ ╲ ¡     + Ï ╲ ë + Ï
C |   , < L \ └ ¿     £ Ü └ ï £ Ü
D |   - = M ] ╱ ▪▪    ' Ñ ╱ ° ' Ñ
E |   . > N ↑ ▬ π     ♣ ♦ ▬ π ♣ ♦
F |   / ? O ─ ┌ ┘     ¨ Ç ┌ Ç ¨ π
```

Table : SPAIN Charset in Uppercase/Graphic Mode (Secondary address = 0)

```
  | 0 1 2 3 4 5 6 7 8 9 a b c d e f
--+
0 |       0 @ p ` P         ← ` P     ←
1 |     ! 1 a q A Q     | à A Q | à
2 |     " 2 b r B R     └ è B R └ è
3 |     # 3 c s C S     ┴ ♠ C S ┴ ♠
4 |     $ 4 d t D T     ┘ ò D T ┘ ò
5 |     % 5 e u E U     ├ ♥ E U ├ ♥
6 |     & 6 f v F V     ─ á F V ─ á
7 |     ' 7 g w G W     ┤ é G W ┤ é
8 |     ( 8 h x H X     ┌ í H X ┌ í
9 |     ) 9 i y I Y     ┬ ó I Y ┬ ó
a |     * : j z J Z     ┐ ú J Z ┐ ú
b |     + ; k [ K ¡     + ï K ¡ + ï
c |     , < l \ L ¿     £ ü L ¿ £ ü
d |     - = m ] M Ñ     ' ñ M Ñ ' ñ
e |     . > n ↑ N π     ♣ ♦ N π ♣ ♦
f |     / ? o ─ O Ç     ¨ ç O Ç ¨ π
```

Table SPAIN Charset in Lowercase/Uppercase Mode (Secondary address = 7)

11.6.6 Sweden

```
  | 0 1 2 3 4 5 6 7 8 9 A B C D E F
--+
0 |       0 @ P ─ ┐         ┌ ─ ┐     ┌
1 |     ! 1 A Q ♠ ●     █ ┴ ♠ ● █ ┴
2 |     " 2 B R | ─     ■ ┬ | ─ ■ ┬
3 |     # 3 C S ─ ♥     ─ ┤ ─ ♥ ─ ┤
4 |     $ 4 D T ─ |     _ | ─ | _ |
5 |     % 5 E U ─ ╱     | █ ─ ╱ | █
6 |     & 6 F V ─ ×     ▓ ■ ─ × ▓ ■
7 |     ' 7 G W | o     | ─ | o | ─
8 |     ( 8 H X | ♣     ▓ ─ | ♣ ▓ ─
9 |     ) 9 I Y ╲ |     ▛ ■ ╲ | ▛ ■
A |     * : J Z ╲ ♦     | ┘ ╲ ♦ | ┘
B |     + ; K Ä ╱ +     ├ ▪ ╱ + ├ ▪
C |     , < L Ö └ ▒     ▪ ▪ └ ▒ ▪ ▪
D |     - = M Å ╲ |     └ ┘ ╲ | └ ┘
E |     . > N ↑ ╱ π     ┐ ▪ ╱ π ┐ ▪
F |     / ? O ← ┌ ◥     ▬ ▪ ┌ ◥ ▬ π
```

Table : SWEDEN Charset in Uppercase/Graphic Mode (Secondary address = 0)

```
  | 0 1 2 3 4 5 6 7 8 9 a b c d e f
--+--------------------------------
0 |       0 @ p — P       ⌐ — P     ⌐
1 |     ! 1 a q A Q     █ ⊥ A Q █ ⊥
2 |     " 2 b r B R     ▬ ⊤ B R ▬ ⊤
3 |     # 3 c s C S     ─ ⊣ C S ─ ⊣
4 |     $ 4 d t D T     _ │ D T _ │
5 |     % 5 e u E U     │ ▌ E U │ ▌
6 |     & 6 f v F V     ▓ ▐ F V ▓ ▐
7 |     ' 7 g w G W     │ ▔ G W │ ▔
8 |     ( 8 h x H X     ▰ ▀ H X ▰ ▀
9 |     ) 9 i y I Y     ▨ ▄ I Y ▨ ▄
a |     * : j z J Z     │ ↙ J Z │ ↙
b |     + ; k ä K Ä     ├ ▪ K Ä ├ ▪
c |     , < l ö L Ö     ▪ ▪ L Ö ▪ ▪
d |     − = m å M Å     └ ┘ M Å └ ┘
e |     . > n ↑ N ⊠     ┐ ▪ N ⊠ ┐ ▪
f |     / ? o ← O ▨     ▄ ▚ O ▨ ▄ ⊠
```

Table SWEDEN Charset in Lowercase/Uppercase Mode (Secondary address = 7)

11.6.7 Switzerland

```
  | 0 1 2 3 4 5 6 7 8 9 A B C D E F
--+--------------------------------
0 |       0 @ P ` ┐       § ` ┐     §
1 |     ! 1 A Q ◣ ◤     │ à ◣ ◤ │ à
2 |     " 2 B R ▬ ╱     └ è ▬ ╱ └ è
3 |     # 3 C S ▰ ▪     ⊥ ì ▰ ▪ ⊥ ì
4 |     $ 4 D T ▪ ╲     ┘ ò ▪ ╲ ┘ ò
5 |     % 5 E U ▪ ▌     ├ ù ▪ ▌ ├ ù
6 |     & 6 F V ╲ ▬     ─ â ╲ ▬ ─ â
7 |     ' 7 G W ╱ ▪     ┤ ê ╱ ▪ ┤ ê
8 |     ( 8 H X ▰ ▬     ⌐ î ▰ ▬ ⌐ î
9 |     ) 9 I Y │ _     ⊤ ô │ _ ⊤ ô
A |     * : J Z │ ▌     ┐ û │ ▌ ┐ û
B |     + ; K [ ╲ ë     + Ä ╲ ë + Ä
C |     , < L \ └ ï     £ Ö └ ï £ Ö
D |     − = M ] ╱ ²     ' Ü ╱ ² ' Ü
E |     . > N ↑ ─ π     ^ ß ─ π ^ ß
F |     / ? O — ⌐ ç     ¨ é ⌐ ç ¨ π
```

Table : SWITZERLAND Charset in Uppercase/Graphic Mode (Secondary address = 0)

```
  | 0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
--+--------------------------------------------------
0 |          0  @  p  `  P              §  `  P     §
1 |       !  1  a  q  A  Q        |  à  A  Q  |  à
2 |       "  2  b  r  B  R        └  è  B  R  └  è
3 |       #  3  c  s  C  S        ┴  ì  C  S  ┴  ì
4 |       $  4  d  t  D  T        ┘  ò  D  T  ┘  ò
5 |       %  5  e  u  E  U        ├  ù  E  U  ├  ù
6 |       &  6  f  v  F  V        ─  â  F  V  ─  â
7 |       '  7  g  w  G  W        ┤  ê  G  W  ┤  ê
8 |       (  8  h  x  H  X        ┌  î  H  X  ┌  î
9 |       )  9  i  y  I  Y        ┬  ô  I  Y  ┬  ô
a |       *  :  j  z  J  Z        ┐  û  J  Z  ┐  û
b |       +  ;  k  [  K  ë        +  ä  K  ë  +  ä
c |       ,  <  l  \  L  ï        £  ö  L  ï  £  ö
d |       -  =  m  ]  M  ²        '  ü  M  ²  '  ü
e |       .  >  n  ↑  N  π        ^  ß  N  π  ^  ß
f |       /  ?  o  ─  O  ç        ¨  é  O  ç  ¨  π
```

Table SWITZERLAND Charset in Lowercase/Uppercase Mode (Secondary address = 7)

11.7 EPSON FX-80 character table

11.7.1 Basic charset

```
   O  1  2  3  4  5  6  7  8  9  A  B  C  D  E  F
0           0  @  P  `  p              0  @  P  `  p
1        !  1  A  Q  a  q        /  1  A  Q  a  q
2        "  2  B  R  b  r        "  2  B  R  b  r
3        #  3  C  S  c  s        #  3  C  S  c  s
4        $  4  D  T  d  t        $  4  D  T  d  t
5        %  5  E  U  e  u        %  5  E  U  e  u
6        &  6  F  V  f  v        &  6  F  V  f  v
7        '  7  G  W  g  w        '  7  G  W  g  w
8        (  8  H  X  h  x        (  8  H  X  h  x
9        )  9  I  Y  i  y        )  9  I  Y  i  y
A        *  :  J  Z  j  z        *  :  J  Z  j  z
B        +  ;  K  [  k  {        +  ;  K  [  k  {
C        ,  <  L  \  l  |        ,  <  L  \  l  /
D        -  =  M  ]  m  }        -  =  M  ]  m  }
E        .  >  N  ^  n  ~        .  >  N  ^  n  ~
F        /  ?  O  _  o           /  ?  O  _  o
```

11.7.2 Extended charset

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	à	§		0	@	P	`	p	à	§		O	@	P	`	p
1	è	ß	!	1	A	Q	a	q	è	ß	/	1	A	Q	a	q
2	ù		"	2	B	R	b	r	ù		"	2	B	R	b	r
3	ò		#	3	C	S	c	s	ò		#	3	C	S	c	s
4	ì		$	4	D	T	d	t	ì		$	4	D	T	d	t
5	°	ø	%	5	E	U	e	u	°	Ø	%	5	E	U	e	u
6	£	¨	&	6	F	V	f	v	£	¨	&	6	F	V	f	v
7		Ä	'	7	G	W	g	w		Ä	'	7	G	W	g	w
8		Ö	(	8	H	X	h	x		Ö	(	8	H	X	h	x
9		Ù	)	9	I	Y	i	y		Ù	)	9	I	Y	i	y
A	ä		*	:	J	Z	j	z	ä		*	:	J	Z	j	z
B			+	;	K	[	k	{			+	;	K	[	k	{
C	ü		,	<	L	\	l	|	ü		,	<	L	\	l	|
D	É		-	=	M	]	m	}	É		-	=	M	]	m	}
E	é		.	>	N	^	n	~	é		.	>	N	^	n	~
F	¥		/	?	O	_	o		¥		/	?	O	_	o	Ø

11.7.3 International charsets changes

CHARSET	23h	24h	40h	5Bh	5ch	5dh	5eh	60h	7Bh	7Ch	7Dh	7Eh
Basic	#	$	@	[	\	]	^	`	{	|	}	~
USA	#	$	@	[	\	]	¨	`	{	|	}	~
France	#	$	à	°	ç	§	^	`	é	ù	è	¨
Germany	#	$	§	Ä	Ö	Ü	^	`	ä	ö	ü	ü
UK	£	$	@	[	\	]	^	`	{	|	}	~
Denmark I	#	$	@	Æ	Ø	Å	^	`	æ	ø	å	~
Sweden	#	¤	É	Ä	Ö	Å	Ü	é	ä	ö	å	ü
Italy	#	$	@	°	\	é	^	ù	à	ò	è	ì
Spain	₧	$	@	¡	Ñ	¿	^	`	¨	ñ	}	~
Japan	#	$	@	[	¥	]	^	`	{	|	}	~
Norway	#	¤	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü
Denmark II	#	$	É	Æ	Ø	Å	Ü	é	æ	ø	å	ü

11.8 IBM character tables

11.8.1 Table 1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	p			á	▓	└	╨	α	≡
1			!	1	A	Q	a	q			í	▓	┴	╤	ß	±
2			"	2	B	R	b	r			ó	▓	┬	╥	Γ	≥
3			#	3	C	S	c	s			ú	│	├	╙	π	≤
4			$	4	D	T	d	t			ñ	┤	─	╘	Σ	⌠
5			%	5	E	U	e	u			Ñ	╡	┼	╒	σ	⌡
6			&	6	F	V	f	v			ª	╢	╞	╓	µ	÷
7			'	7	G	W	g	w			º	╖	╟	╫	τ	≈
8			(	8	H	X	h	x			¿	╕	╚	╪	Φ	°
9			)	9	I	Y	i	y			⌐	╣	╔	┘	Θ	∙
A			*	:	J	Z	j	z			¬	║	╩	┌	Ω	·
B			+	;	K	[	k	{			½	╗	╦	■	δ	√
C			,	<	L	\	l	|			¼	╝	╠	■	∞	ⁿ
D			-	=	M	]	m	}			¡	╜	═	■	ø	²
E			.	>	N	^	n	~			«	╛	╬	■	∈	■
F			/	?	O	_	o				»	┐	╧	■	∩	

11.8.2 Table 2

International 1

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	p	Ç	É	á	▓	└	╨	α	≡
1			!	1	A	Q	a	q	ü	æ	í	▓	┴	╤	ß	±
2			"	2	B	R	b	r	é	Æ	ó	▓	┬	╥	Γ	≥
3	♥		#	3	C	S	c	s	â	ô	ú	│	├	╙	π	≤
4	♦		$	4	D	T	d	t	ä	ö	ñ	┤	─	╘	Σ	⌠
5	♣	§	%	5	E	U	e	u	à	ò	Ñ	╡	┼	╒	σ	⌡
6	♠		&	6	F	V	f	v	å	û	ª	╢	╞	╓	µ	÷
7			'	7	G	W	g	w	ç	ù	º	╖	╟	╫	τ	≈
8			(	8	H	X	h	x	ê	ÿ	¿	╕	╚	╪	Φ	°
9			)	9	I	Y	i	y	ë	Ö	⌐	╣	╔	┘	Θ	∙
A			*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	·
B			+	;	K	[	k	{	ï	¢	½	╗	╦	■	δ	√
C			,	<	L	\	l	|	î	£	¼	╝	╠	■	∞	ⁿ
D			-	=	M	]	m	}	ì	¥	¡	╜	═	■	ø	²
E			.	>	N	^	n	~	Ä	₧	«	╛	╬	■	∈	■
F			/	?	O	_	o		Å	ƒ	»	┐	╧	■	∩	

International 2

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	p	Ç	É	á	▒	└	╨	α	≡
1			!	1	A	Q	a	q	ü	æ	í	▒	┴	╤	β	±
2			"	2	B	R	b	r	é	Æ	ó	▓	┬	╥	Γ	≥
3	♥		#	3	C	S	c	s	â	ô	ú	▓	├	╨	π	≤
4	♦		$	4	D	T	d	t	ä	ö	ñ	┤	─	╘	Σ	⌠
5	♣	§	%	5	E	U	e	u	à	ò	Ñ	╡	┼	╒	σ	⌡
6	♠		&	6	F	V	f	v	å	û	ª	╢	╞	╓	μ	÷
7			'	7	G	W	g	w	ç	ù	º	╖	╟	╫	τ	≈
8			(	8	H	X	h	x	ê	ÿ	¿	╕	╚	╪	Φ	°
9			)	9	I	Y	i	y	ë	Ö	⌐	╣	╔	┘	Θ	∙
A			*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	·
B			+	;	K	[	k	{	ï	¢	½	╗	╦	█	δ	√
C			,	<	L	\	l	|	î	£	¼	╝	╠	▄	∞	ⁿ
D			-	=	M	]	m	}	ì	Ø	¡	╜	═	▌	ø	²
E			.	>	N	^	n	~	Ä	₧	³	╛	╬	▐	∈	■
F			/	?	O	_	o		Å	ƒ	¤	┐	╧	▀	∩	

Israel

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	p	א	ן	á	▒	└	╨	α	≡
1			!	1	A	Q	a	q	ב	ע	í	▒	┴	╤	β	±
2			"	2	B	R	b	r	ג	ף	ó	▓	┬	╥	Γ	≥
3	♥		#	3	C	S	c	s	ד	פ	ú	▓	├	╨	π	≤
4	♦		$	4	D	T	d	t	ה	ץ	ñ	┤	─	╘	Σ	⌠
5	♣	§	%	5	E	U	e	u	ו	צ	Ñ	╡	┼	╒	σ	⌡
6	♠		&	6	F	V	f	v	ז	ק	ª	╢	╞	╓	μ	÷
7			'	7	G	W	g	w	ח	ר	º	╖	╟	╫	τ	≈
8			(	8	H	X	h	x	ט	ש	¿	╕	╚	╪	Φ	°
9			)	9	I	Y	i	y	י	ת	⌐	╣	╔	┘	Θ	∙
A			*	:	J	Z	j	z	ך	∩	¬	║	╩	┌	Ω	·
B			+	;	K	[	k	{	כ	¢	½	╗	╦	█	δ	√
C			,	<	L	\	l	|	ל	Ψ	¼	╝	╠	▄	∞	ⁿ
D			-	=	M	]	m	}	ם	¥	¡	╜	═	▌	ø	²
E			.	>	N	^	n	~	מ	₧	«	╛	╬	▐	∈	■
F			/	?	O	_	o		ן	ƒ	»	┐	╧	▀	∩	

Greece

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	@	P	`	p	A	P	ι	⫶	∟	╨	ω	Ω
1			!	1	A	Q	a	q	B	Σ	κ		⊥	╤	ά	Ω
2			"	2	B	R	b	r	Γ	T	λ		┬	Ⅱ	έ	≥
3	♥		#	3	C	S	c	s	Δ	Y	μ		├	Ⅱ	ή	≤
4	♦		$	4	D	T	d	t	E	Φ	ν		─	⊨	ϊ	∫
5	♣	§	%	5	E	U	e	u	Z	X	ξ		┼	⌐	(	∫
6	♠		&	6	F	V	f	v	H	Ψ	o		╞	⌐	ό	÷
7			'	7	G	W	g	w	Θ	Ω	π	η	┠	╫	ύ	≈
8			(	8	H	X	h	x	I	α	ρ	╡	┠	╪	ϋ	°
9			)	9	I	Y	i	y	K	β	σ		╟		ώ	£
A			*	:	J	Z	j	z	Λ	γ	ς			╒	Ά	·
B			+	;	K	[	k	(	M	δ	τ		╥	█	Έ	√
C			,	<	L	\	l	¦	N	ε	υ	]	╠	█	Ή	ⁿ
D			-	=	M	]	m	)	Ξ	ζ	φ	╜	╟	█	Ί	²
E			.	>	N	^	n	~	O	η	χ	╛	╫	█	Ό	■
F			/	?	O	_	o	Π	ϑ	ψ			╧	█	Ύ	

Portugal

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0			0	@	P	`	p	Ç	É	á	⫶	∟	╨	α	≡
1			!	1	A	Q	a	q	ü	æ	í		⊥	╤	β	±
2			"	2	B	R	b	r	é	Æ	ó		┬	Ⅱ	Γ	≥
3	♥		#	3	C	S	c	s	â	ô	ú		├	Ⅱ	π	≤
4	♦		$	4	D	T	d	t	ä	ö	ñ		─	⊨	Σ	∫
5	♣	§	%	5	E	U	e	u	à	ò	Ñ		┼	⌐	σ	∫
6	♠		&	6	F	V	f	v	å	û	ª		╞	⌐	μ	÷
7			'	7	G	W	g	w	ç	ù	º	η	┠	╫	τ	≈
8			(	8	H	X	h	x	ê	ÿ	¿	╡	┠	╪	Φ	°
9			)	9	I	Y	i	y	ë	Ö	⌐		╟		Θ	·
A			*	:	J	Z	j	z	è	Ü	¬			╒	Ω	·
B			+	;	K	[	k	(	ï	¢	Ú		╥	█	δ	√
C			,	<	L	\	l	¦	î	£	ƒ	]	╠	█	∞	ⁿ
D			-	=	M	]	m	)	ì	À	¡	╜	╟	█	Ø	²
E			.	>	N	^	n	~	Ä	Ê	³	╛	╫	█	∈	■
F			/	?	O	_	o	Å	Ô	ó			╧	█	∩	

Spain

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0				0	@	P	`	p	Ç	É	á	▦	└	╨	α	≡
1			!	1	A	Q	a	q	ü	f	í	▦	⊥	╤	ß	±
2			"	2	B	R	b	r	é	ó	ó	▦	┬	╥	Γ	≥
3	♥		#	3	C	S	c	s	â	ô	ú	│	├	╨	π	≤
4	♦		$	4	D	T	d	t	ä	ö	ñ	┤	─	╘	Σ	⌠
5	♣	§	%	5	E	U	e	u	à	ò	Ñ	╡	┼	╒	σ	⌡
6	♠		&	6	F	V	f	v	Â	û	ª	╢	╞	╓	μ	÷
7			'	7	G	W	g	w	ç	ù	º	╖	╟	╫	τ	≈
8			(	8	H	X	h	x	ê	Â	¿	╕	╚	╪	Φ	°
9			)	9	I	Y	i	y	ë	Ö	⌐	╣	╔	┘	Θ	∙
A			*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	·
B			+	;	K	[	k	{	ï	ò	½	╗	╦	█	δ	√
C			,	<	L	\	l	|	î	£	¼	╝	╠	▄	∞	ⁿ
D			-	=	M	]	m	}	ì	ú	¡	╜	═	▌	ø	²
E			.	>	N	^	n	~	Ä	R	«	╛	╬	▐	€	■
F			/	?	O	_	o		È	Ÿ	»	┐	╧	▀	∩	

11.9 Technical Specifications

Output Type PNG file 2-bit depth (4 grey levels) with lossless compression using LodePNG written by Lode Vandevenne (http://lodev.org/lodepng/)

typical file size range is 30kB - 140kB

Page size 1984 x 2580

Printable area size 1920 x 2160 (80 PICA characters and 60 lines at 1/6")

Horizontal Resolution 240 dpi

Vertical Resolution 216 dpi

Physical ratio A4 (21cm x 29,7cm)

Character matrix 8V x 11H in draft mode
16V x 12H in NLQ mode

Print pitches Pica, 10 char/in, 80 char/line
Elite, 12 char/in, 96 char/line
Micro, 15 char/in, 120 char/line
Condensed, 17.1 char/in, 137 char/line
Pica Compressed, 20 char/in, 160 char/line
Elite Compressed, 24 char/in, 192 char/line
Micro Compressed, 30 char/in, 240 char/line

Printing styles Boldface
Double width

Superscript
Subscript
Double strike
Underlined
Italic
Reversed
Overlined

11.10 Print Sample

With Printer Ink Density set to Medium. Emulation is Commodore MPS.

MPS EMULATION PRINT TEST PAGE

```
DRAFT Simple    Under,Agqp  Bold  Super  sub  Rev
ITALIC Simple   Under,Agqp  Bold  Super  sub  Rev
NLQ Simple      Under,Agqp  Bold  Super  sub  Rev

DRAFT Double    Under,Agqp  Bold  Super  sub  Rev
ITALIC Double   Under,Agqp  Bold  Super  sub  Rev
NLQ Double      Under,Agqp  Bold  Super  sub  Rev

DRAFT Large     Under,Agqp  Bold  Super  sub  Rev
ITALIC Large    Under,Agqp  Bold  Super  sub  Rev
NLQ Large       Under,Agqp  Bold  Super  sub  Rev

DRAFT Lg Db     Under,Agqp  Bold  Super  sub  Rev
ITALIC Lg Db    Under,Agqp  Bold  Super  sub  Rev
NLQ Lg Db       Under,Agqp  Bold  Super  sub  Rev
```

```
PICA                Draft Regular    Draft Italic    Near Letter Quality
ELITE               Draft Regular    Draft Italic    Near Letter Quality
MICRO               Draft Regular    Draft Italic    Near Letter Quality
CONDENSED           Draft Regular    Draft Italic    Near Letter Quality
PICA COMPRESSED     Draft Regular    Draft Italic    Near Letter Quality
ELITE COMPRESSED    Draft Regular    Draft Italic    Near Letter Quality
MICRO COMPRESSED    Draft Regular    Draft Italic    Near Letter Quality
```

```
GRAPHIC BITMAP
Simple Bitmap

     **** COMMODORE 64 BASIC V2 ****
 64K RAM SYSTEM  38911 BASIC BYTES FREE
READY.
▉

Repeated Bitmap
```

```
PETASCII code tables

    UPPER/GRAPHIC                          LOWER/UPPER

 | 0 1 2 3 4 5 6 7 8 9 A B C D E F      | 0 1 2 3 4 5 6 7 8 9 A B C D E F
-+-------------------------------      -+-------------------------------
0|       0 @ P ─ ┐       ┌ ─ ┐  ┌     0|       0 @ p ─ P       ┌ ─ P  ┌
1|   ! 1 A Q ♠ ●    ▌ ┴ ♠ ● ▌ ┴      1|   ! 1 a q A Q    ▌ ┴ A Q ▌ ┴
2|   " 2 B R │ ─    ▬ ┬ │ ─ ▬ ┬      2|   " 2 b r B R    ▬ ┬ B R ▬ ┬
3|   # 3 C S ─ ♥    ─ ┤ ─ ♥ ─ ┤      3|   # 3 c s C S    ─ ┤ C S ─ ┤
4|   $ 4 D T ─ │    ─ │ ─ │ ─ │      4|   $ 4 d t D T    ─ │ D T ─ │
5|   % 5 E U ─ ╱    │ │ ─ ╱ │ │      5|   % 5 e u E U    │ │ E U │ │
6|   & 6 F V ─ ×    ▓ ▌ ─ × ▓ ▌      6|   & 6 f v F V    ▓ ▌ F V ▓ ▌
7|   ' 7 G W │ ○    │ ─ │ ○ │ ─      7|   ' 7 g w G W    │ ─ G W │ ─
8|   ( 8 H X │ ♣    ▒ ─ │ ♣ ▒ ─      8|   ( 8 h x H X    ▒ ─ H X ▒ ─
9|   ) 9 I Y ╲ │    ◤ ▄ ╲ │ ◤ ▄      9|   ) 9 i y I Y    ╱ ▄ I Y ╱ ▄
A|   * : J Z ╲ ♦    │ │ ╲ ♦ │ ┘      A|   * : j z J Z    │ ╭ J Z │ ┘
B|   + ; K [ ╱ +    ├ ╽ ╱ + ├ ▪      B|   + ; k [ K +    ├ ▪ K + ├ ▪
C|   , < L £ ┗ ▒    ▪ ▪ ┗ ▒ ▪ ▪      C|   , < l £ L ▒    ▪ ▪ L ▒ ▪ ▪
D|   - = M ] ╲ │    ┗ ┘ ╲ │ ┗ ┘      D|   - = m ] M │    ┗ ┘ M │ ┗ ┘
E|   . > N ↑ ╱ π    ┐ ▪ ╱ π ┐ ▪      E|   . > n ↑ N ▩    ┐ ▪ N ▩ ┐ ▪
F|   / ? O ← ┌ ▜    ▃ ▪ ┌ ▜ ▃ π      F|   / ? o ← O ▨    ▃ ▪ O ▨ ▃ ▩
```

11.11 Document Revisions

Revi-sion	Date	Author	Description
1.0.0	May 27, 2016	René Garcia	Initial release
1.0.1	May 30, 2016	René Garcia	Corrected capabilities table and options BIT IMG SUB corrected Ink Density samples
1.1	February 18, 2018	René Garcia	Rename MPS Printer Emulation to Virtual Printer New feature: ASCII output format

REST API CALLS

Starting from Ultimate firmware 3.11, the application supports API calls by means of the HTTP protocol. This simplifies the integration of remote control functionality in external applications, as it follows a well defined standard.

The format of the URL is as follows:

`/v1/<route>/<path>:<command>?<arguments>`

The 'verb' depends on the kind of operation. In general:

Table 1: URL Verbs

Verb	Meaning
GET	Retrieves information without changing state
PUT	Sends information, or performs an action, using the information in the URL or in a file referenced by the URL.
POST	Perform an action using the information that is attached to the request.

What is returned with the request depends on the command. In most cases the command returns a valid JSON message, using the *Content-Type: application/json* string in the header. The JSON contains at least one entry, called *errors*. This is a list (array) of strings with things that went wrong during the execution of the command. A complete response could, for instance, look like this:

```
HTTP/1.1 200 OK
Connection: close
Content-Type: application/json
Content-Length: 22

{
  "errors": []
}
```

12.1 Routes

12.1.1 About

URL	Param-eters	Action

GET /v1/version		Returns the current version of the ReST API: .. code-block:
		
		{
		    "version": "0.1",
		    "errors": []
		}

12.1.2 Runners

URL	Parameters	Action
PUT /v1/ runners:sidplay	*file* [songnr]	This command requests the Ultimate to play a SID file. The *file* argument points to an existing file in the file system of the Ultimate. It plays the default song, unless the optional *songnr* argument is given. The player will attempt to open the song lengths file in the subdirectory 'SONGLENGTHS'.
POST /v1/ runners:sidplay	[songnr]	This command requests the Ultimate to play a SID file that is attached as a file to the request. It plays the default song, unless the optional *songnr* argument is given. An optional second attachment can be sent that contains the song lengths.
PUT /v1/ runners:modplay	*file*	This command requests the Ultimate to play an Amiga MOD file. The *file* argument points to an existing file in the file system of the Ultimate.
POST /v1/ runners:modplay		This command requests the Ultimate to play the Amiga MOD file that is attached as a file to the request.
PUT /v1/ runners:load_prg	*file*	With this command a progam can be loaded into memory. The *file* argument points to an existing file in the file system of the Ultimate. The machine resets, and loads the designated program into memory using DMA. It does not automatically run the program.
POST /v1/ runners:load_prg		With this command a progam can be loaded into memory. The machine resets, and loads the attached program into memory using DMA. It does not automatically run the program.
PUT /v1/ runners:run_prg	*file*	With this command a progam can be loaded into memory. The *file* argument points to an existing file in the file system of the Ultimate. The machine resets, and loads the designated program into memory using DMA. Then it automatically runs the program.
POST /v1/ runners:run_prg		With this command a progam can be loaded into memory. The machine resets, and loads the attached program into memory using DMA. Then it automatically runs the program.
PUT /v1/ runners:run_crt	*file*	With this command a cartridge file can be started. The *file* argument points to an existing file in the file system of the Ultimate. The machine resets, with the specified cartridge active. It does not alter the configuration of the Ultimate.
POST /v1/ runners:run_crt		This command starts a supplied cartridge file. The 'crt' file is attached to the POST request. The machine resets, with the attached cartridge active. It does not alter the configuration of the Ultimate.

12.1.3 Configuration

URL	Parameters	Action

`GET /v1/configs`		This command obtains a list of all the configuration categories in the Ultimate application. Note that there is no path specified after 'configs'. An example result is: .. code-block:
		
		```HTTP/1.1 200 OK```
		```Connection: close```
		```Content-Type: application/json```
		```Content-Length: 405```
		
		```{```
		``` "categories": [```
		``` "Audio Mixer",```
		``` "SID Sockets Configuration",```
		``` "UltiSID Configuration",```
		``` "SID Addressing",```
		``` "C64 and Cartridge Settings",```
		``` "U64 Specific Settings",```
		``` "Clock Settings",```
		``` "Network settings",```
		``` "WiFi settings",```
		``` "Modem Settings",```
		``` "LED Strip Settings",```
		``` "Data Streams",```
		``` "Software IEC Settings",```
		``` "User Interface Settings",```
		``` "Tape Settings",```
		``` "Drive A Settings",```
		``` "Drive B Settings"```
		```],```
		``` "errors": []```
		```}```

`GET /v1/configs/`
`<category>`
		This command obtains a list of all the configuration items in the category specified in the URL. Wildcards are allowed. Note that the depth of the specified path is 1. It specifies the category. Example: `GET /v1/configs/drive%20a*` results in: .. code-block:
		
		```HTTP/1.1 200 OK```
		```Connection: close```
		```Content-Type: application/json```
		```Content-Length: 414```
		
		```{```
		``` "Drive A Settings": {```
		``` "Drive": "Enabled",```
		``` "Drive Type": "1541",```
		``` "Drive Bus ID": 8,```
		``` "ROM for 1541 mode": "1541.rom",```
		``` "ROM for 1571 mode": "1571.rom",```
		``` "ROM for 1581 mode": "1581.rom",```
		``` "Extra RAM": "Disabled",```
		``` "Disk swap delay": 1,```
		``` "Resets when C64 resets": "Yes",```
		``` "Freezes in menu": "Yes",```
		``` "GCR Save Align Tracks": "Yes",```
		``` "Leave Menu on Mount": "Yes"```
		``` },```

12.1. Routes

12.1.4 Machine

URL	Parameters	Action
PUT /v1/ machine:reset		This command sends a reset to the machine. The current configuration is not changed.
PUT /v1/ machine:reboot		This command restarts the machine. It re-initializes the cartridge configuration and sends a reset to the machine.
PUT /v1/ machine:pause		When issuing this command, the machine is paused by pulling the DMA line low at a safe moment. This stops the CPU. Note that this does not stop any timers.
PUT /v1/ machine:resume		With this command, the machine is resumed from the paused state. The DMA line is released and the CPU will continue where it left off.
PUT /v1/ machine:poweroff		This U64-only command causes the machine to power off. Note that it is likely that you won't receive a valid response.
PUT /v1/ machine:writemem	*address* *data*	With this command, data can be written to C64 memory. To be more exact: this command writes data through DMA, so the memory map that is currently selected is used. Writing to the I/O registers of the 6510 is not possible. Data bytes are written in consequetive memory locations. The *address* argument specifies the memory location in hexadecimal format. The *data* argument contains a string of bytes in hexadecimal format. The maxmimum number of bytes written with this method is 128. Example: PUT /v1/machine:writemem?address=D020&data=0504 This results in 05 being written to \$D020 and 04 being written to \$D021. In other words: the border will be green and the main screen will turn purple.
POST /v1/ machine:writemem	*address*	With this command, data can be written to C64 memory. The data, passed as a binary attachment, will be written to memory starting from the location indicated by the *address* argument, which shall be formatted in hexadecimal. The data should not wrap around \$FFFF.
GET /v1/ machine:readmem	*address* *[length]*	This command performs a DMA read action on the cartridge bus and returns the result as a binary attachment. The *address* argument specifies the memory location in hexadecimal format. The optional argument *length* specifies the number of bytes being read. When not specified, 256 bytes are returned.
GET /v1/ machine:debugreg		This command reads the debug register (\$D7FF) and returns it in the "value" field of the JSON response. The value is in hexadecimal format. *This is currently an U64-only call.*
PUT /v1/ machine:debugreg	*value*	This command writes the value specified by the *value* argument (in hexadecimal) into the debug register (\$D7FF), and then reads the debug register (\$D7FF) and returns it in the "value" field of the JSON response. *This is currently an U64-only call.*

12.1.5 Floppy Drives

URL	Parameters	Action
`GET /v1/drives`		With this command, the information about all the (internal) drives on the IEC bus is returned. In addition to the presence, it also shows the image files and paths of the mounted disks or referenced paths. An example follows: .. code-block:

```
{
    "drives":[
        {
            "a":{
                "enabled":true,
                "bus_id":8,
                "type":"1581",
                "rom":"1581.rom",
                "image_file":"",
                "image_path":""
            }
        },
        {
            "b":{
                "enabled":false,
                "bus_id":9,
                "type":"1541",
                "rom":"1541.rom",
                "image_file":"",
                "image_path":""
            }
        },
        {
            "softiec":{
                "enabled":false,
                "bus_id":11,
                "type":"DOS emulation",
                "last_error":"73,U64IEC ULTIMATE DOS V1.1,
→00,00",
                "partitions":[
                    {
                        "id":0,
                        "path":"/Temp/"
                    }
                ]
            }
        }
    ],
    "errors":[

    ]
}
```

URL	Parameters	Action
`PUT /v1/drives/ <drive>:mount`	*image* *[type]* *[mode]*	This command can be used to mount an existing image onto the drive specified in the path. The *image* argument points to the file in the file system of the Ultimate. The optional *type* argument specifies the type of the image, and could be one of the following: **d64**, **g64**, **d71**, **g71** or **d81**. If this argument is omitted, it will use the file extension of the file. The optional *mode* argument can be one of the following: **readwrite**, **readonly** or **unlinked**. In *readwrite* mode, the drive can write to the image file; in *readonly* mode the disk is write protected and in *unlinked* mode the disk is

12.1.6 Data Streams (U64 only)

The U64 supports streaming video and audio over its LAN port. The following API commands are available to control these streams.

URL	Parameters	Action
PUT /v1/streams/ <stream name>:start	*ip*	Use this command to start one of the available streams. Valid stream names are **video**, **audio** and **debug**. The IP number parameter is required for the U64 to know where to send the stream to. The default port number that the data stream is sent to is 11000 for the video stream, 11001 for the audio stream and 11002 for the debug stream. A custom port number can be added to the IP address, after a colon separator; e.g. *192.168.178.224:6789* . Note that turning on the video stream will automatically turn off the debug stream.
PUT /v1/streams/ <stream name>:stop		With this command a data stream can be turned off. Valid stream names are **video**, **audio** and **debug**.

12.1.7 File Manipulation

This section lists the API commands for file manipulation. This is the current state of V3.11 alpha. It is not yet finished.

URL	Parameters	Action
GET /v1/files/ <path>:info		This command returns basic information about a file, like size and extension. It simply performs an 'fstat'. Supports wildcards. *Unfinished*
PUT /v1/files/ <path>:create_d64	*[tracks]* *[diskname]*	With this command a .d64 file can be created. The full path shall be specified from the root of the file system, including the file to be created. The default number of tracks is 35, but it can also be set to 40. The optional *diskname* argument overrides the name to be used in the header of the disk. When not given, it is taken from the name of the file that is being created.
PUT /v1/files/ <path>:create_d71	*[diskname]*	With this command a .d71 file can be created. The full path shall be specified from the root of the file system, including the file to be created. The number of tracks is fixed at 70. The optional *diskname* argument overrides the name to be used in the header of the disk. When not given, it is taken from the name of the file that is being created.
PUT /v1/files/ <path>:create_d81	*[diskname]*	With this command a .d81 file can be created. The full path shall be specified from the root of the file system, including the file to be created. The number of tracks is fixed at 160 (80 on each side). The optional *diskname* argument overrides the name to be used in the header of the disk. When not given, it is taken from the name of the file that is being created.
PUT /v1/files/ <path>:create_dnp	*tracks* *[diskname]*	With this command a .dnp file can be created. The full path shall be specified from the root of the file system, including the file to be created. The number of tracks is a required argument to this function. Each track will have 256 sectors. The maximum number of tracks is 255, which makes the maximum DNP size almost 16 Megabytes. The optional *diskname* argument overrides the name to be used in the header of the disk. When not given, it is taken from the name of the file that is being created.

ASSEMBLY 64 INTEGRATION

Starting from firmware 3.11, the Ultimate integrates support for the Assembly64 online database. Because this database is very large, finding titles would be difficult if it was organized as a directory structure. Therefore, the available software can be found using a database search query.

In the main screen, press F6 to open the search query; OR, press F5 and choose "New Search" from the action menu. If a valid network link exists, the following screen appears:

```
** Ultimate 64 Elite V1.43 - 3.11 **

        Assembly 64 Query Form
Name:        _____
Group:       _____
Handle:      _____
Event:       _____
Date:        _____
Rating:      _____
Type:        _____
Order:       _____
Sort:        _____
Latest:      _____
Repo:        _____
Category:    _____
Subcat:      _____

        <<   Search   >>


/                         -F3=Help-
```

13.1 Filling out the Query Form

This screen allows you to enter the query in an effective manner. The first four lines are string fields. You can simply start typing when the line is selected. A cursor will appear and the current line will become an edit field. When done, simply press return.

The subsequent lines are the so called 'presets'. These presets are dynamic and are part of the database structure. It might therefore happen that the screen or options will change over time without a firmware update. The presets can be chosen from the context menu. Hit RETURN, or type the first letter of an item in the drop down list. Press RETURN to select.

Once you have filled out the query form to your liking, move down to the search button and press RETURN:

13.2 First Results Page

The first page lists the first 20 results from the query. At this point (firmware V3.11) it is not possible to scroll to obtain more results. You'd have to narrow down your search query instead.

```
  ** Ultimate 64 Elite V1.43 - 3.11 **

Jumpman Junior (Wattenscheider Crackin
Jumpman Junior (N.O.W.S.E, 1989)
Jumpman + (Cracker Force Nijmegen)
Jumpman Junior + (Cracker Force Nijmeg
Jumpman Junior + (Lion Heart, 1985)
Jumpman Junior + (RCS International, 1
Jumpman Junior + (The Ultimate Mayas)
Jumpman Junior (Feuerteufel)
Jumpman Junior (Mr. 99, 1984)
Jumpman Junior
Jumpman + (Hulkamania)
Jumpman Junior + (Greg)
Jumpman Junior Extended (Oleander, 198
Jump Man BASIC
Jumpman
Jumpman Junior
Jumpman II
jumpman (epyx)
jumpman junior (epyx)
Jumpman (Derbyshire Ram, 1989)

/                              -F3=Help-
```

The results on this page are called "entries". Entires are a collection of one or more files that make up this title. For example, games or demos could consist of multiple "D64" disk images. In this case, all these disk images are part of an entry. In this screen, the entry is shown as one single line.

To see the indiviual files of an entry, navigate to it and press CRSR-RIGHT or RETURN.

13.3 Second Results Page

Once an entry has been selected, the second page with results show the individual files that belong to the entry. In this example, the title consists of just one disk:

```
  ** Ultimate 64 Elite V1.43 - 3.11 **

jumpman junior-wcs.d64        D64  171K




/a64/237033/0/                 -F3=Help-
```

This screen works somewhat like a directory on a USB stick. The file types are recognized and the context menus show the operations that can be done on such a file.
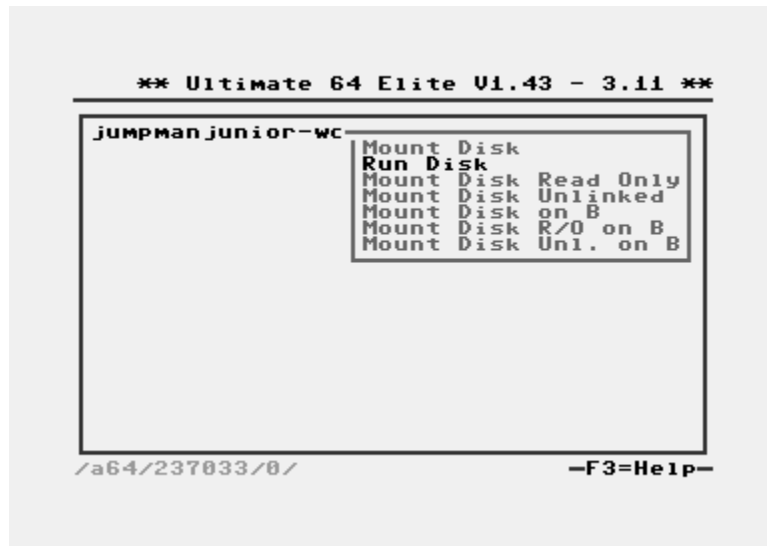
When 'Run Disk' is selected, as in this example, the D64 file will be downloaded and cached in the /Temp folder of the virtual file system of the Ultimate. From there, it will be mounted as usual. The download will take a few seconds, during which you will have to wait patiently. Once the title has been loaded, the title will be loaded:

From there: Enjoy the game!

Note: With the introduction of firmware V3.11, the /Temp folder is not automatically cleaned up. This means that after repeatingly downloading titles, it is possible that the /Temp folder is full and further downloads will fail. This needs to be fixed at some point.

13.4 Contribution

Assembly64 is hosted on a server in Sweden. Note that there are costs incurred with hosting; the servers, the network connection and power are not free. When you like Assembly64, please consider donating through Paypal, using this link:

Donate to Assembly64

FOURTEEN

INTERVIEW RETROMAGAZINE

Interview with Gideon Zweijtzer, creator of the 1541 Ultimate series of expansion cartridges and the Commodore 64 FPGA-based motherboard Ultimate64 - July 2018

Hi, Gideon and thank you so much for accepting my invitation for an interview. All the readers and the editorial staff of RetroMagazine are very excited to have the opportunity to ask you some questions about your experience in designing one of the most (if not the most) famous cartridge/expansion for the C64: the 1541 Ultimate. During 2017, according to your web site, the final steps of the Ultimate64 design have been completed, so the long-awaited board has finally got into production and the first batches have been shipped to the final users earlier this year.

Most of the Commodore 64 fans out there are well-aware of your fantastic products, but I'm pretty sure they don't know how it all began. So let's start from the beginning.

Can you please shortly introduce yourself and tell us something about your own story (ie. where you are born, growing up, your education, your personal interests, etc.)?

Hi David, thanks for the invitation! Talking about myself? Sure... I was born in Amsterdam in 1974, in a quite stable family with one older brother. I have always been interested in technicalities. Before the home computers came, I was always with my technical Lego, although I also loved to race around on my bike through the neighborhood. I often played with circuits made from switches, motors and light bulbs, but unfortunately I did not have anyone in my surroundings with knowledge of electronics. From the secondary school, I went to the university TU Delft, where I studied Electronic Engineering.

I guess you have always been a computer fan and user since when you were a kid. What started you on the path of computing and what was your first experience with a computer? Was the Commodore 64 your fist computer?

I was pretty young when we got an Atari 2600 game console. It was actually my brother who had started with the whole computer thing and he started to investigate the possibilities for programming. There was a basic interpreter for the A2600 at that time, but in the end he bought a ZX81. I was not really allowed to touch it, but sometimes I sneaked into his room and tried a few things, but as a kid without any help, I didn't get that far. Later, my brother got a Commodore 64 and this got big. It was so popular in that time! Computer clubs, meetings, copying games and programs! My brother infected me with his curiosity about programming and although he didn't want me to bother him, I could sit on the floor in between his massive desk and the old color TV that was on top of another table in front of that. As long as he didn't hear me, I could just watch what he was doing. I saw basic, assembler, etc. At a certain point I could tell him from behind the desk that he forgot a statement... It was not until I reached the age of 11 that I got my own Commodore 64.

How did you get started working on a C64, beyond playing games? Did you quickly find interest in programming and discovering how the machine intimately work?

I never really played a lot of games, actually. There are some exceptions, like Giana Sisters. But in general, I did not spend a lot of time on games altogether. As my brother focused a lot on the software, my interest in the hardware grew. At a certain point I made a simple thermometer, using an NTC thermistor on the paddle port of the C64. I needed my older brother again for his math skills to figure out the conversion curve. At the computer club in Amsterdam, I usually spent my time around the repair stand, where I could see how some guys were de-soldering and replacing

chips in broken C64's. According to my mother, I had the full schematic of the C64 hanging from the wall in my small bedroom. But in all honesty, I don't remember that.

After the C64, did you get your first PC and still keep the C64 on your desk? Did you ever use one of the many SD2IEC devices on the market before starting to design the 1541 Ultimate?

No, I haven't. In fact, I think you're now skipping quite a few years. My interest in the C64 faded as the Amiga 500 came, and later the PC. Actually my first PC was a Pentium 120 MHz, so you can imagine that I have resisted PCs for quite some time... The love for the C64 never really went away, I just never used it. Neither have I ever been part of a demo- or game coder group, or the "scene" in general... So there was basically never a need for an SD2IEC or any other C64 peripheral.

What inspired you to design the first version of the 1541 Ultimate and when did you start?

The first version of the 1541 Ultimate was made in 2007. It all started with some implementations of the 6502 as I was learning and getting more experience with FPGA design using VHDL. That was back in 2001 or so. There were a lot of things going on back then. For instance, Jeri Ellsworth was working on her C-One, which later became the DTV, if I am not mistaken. In any case, I had already done a lot of the C64 in FPGA at that time, but I didn't see the point of being a "me-too" player. So I thought I'd do the floppy drive instead. On one of the club meetings that we have in Maarssen, I demoed the very first prototype on a Xilinx Spartan 3 board. You needed a laptop or PC to download a floppy image over Ethernet into its memory, after which the board acted as a floppy drive. No menu, no other emulations, only the drive. Later, in a conversation with one of my colleagues at work, the idea arose to build it into a cartridge, such that the VIC could be used to display a user-interface. This idea crystalized in 2007.

Did you design the hardware and the software/firmware for the 1541 Ultimate all by yourself?

Yes, basically. There have been some important contributions from others over the years, though. But in essence, the hardware design, the FPGA design and the firmware design and framework are made by me.

What was your computer system setup that you used to develop and test the early project of the cartridge?

Just a PC and one Commodore 64... And yes, that did not include a 1541 drive! Later it showed that this was not enough, but I did not have more hardware, so I visited some friends from the Commodore club that had impressive collections of machines to test the compatibility with. In fact, there I found out that the very first prototype of the 1541 Ultimate as a cartridge was not very compatible, which caused some design changes before the board went into production.

Did you take any computer courses to start you in the field of electronics? And if so, what were they and how much time did you invest? Or, like many designers/programmers of the early Eighties, were you a self-taught techie?

Many things were self-taught, although studying at Delft University of Technology has made me understand many more things. But to be fair, I think that I learned most at the job after my studies. I started to work as a junior designer at Technolution B.V., and there I learned most of the practical knowledge that I have today, in terms of electronics design. Interestingly, I brought knowledge about FPGA design back as I was one of the founders of this discipline within the company.

What was your development process like? Did you use to sketch out concepts, design the mainboard and the firmware, etc.? Do you still take on the design process the same way?

Oww, that's quite a difficult question. Because I have always seen these activities as a hobby, I mostly just let it happen. I am the kind of designer that does a lot of design work 'as a background process'. I am not a very structured, method-following, step-by-step kind of engineer. (I made quite a few project-managers pull their hairs out, as they didn't see me work on new tasks they assigned to me in the first weeks...)

I work with iterations, basically. But mostly just in my mind. Sometimes under the shower, or while driving. Once it 'feels right', I start to do some implementation. And sometimes after an implementation I realize it doesn't feel as right anymore. I am not afraid of just throwing some work away and start anew. Of course, always taking into account the lessons learned in the previous step.

Talking about your biggest projects (the 1541 Ultimate cartridge and the new Ultimate64 mainboard), what technical challenge gave you the biggest feeling of accomplishment?

Ok, when I need to limit it to 'technical challenges', it would definitely be the solving of very hard-to-find bugs... You know, those nasty ones that make others quit on their project... Those! On a second place, it is when I power up a new board and everything works right away. (And that's not uncommon in my case... [smug face]).

What was the biggest tech/programming obstacle that you ever overcome while designing/producing/testing/selling the 1541 Ultimate or the Ultimate64?

Obstacles... [thinking]... It depends a bit on how you define the obstacles. Most things are just time consuming tasks. But yet, I think there are several 'obstacles'. I think in case of the 1541 Ultimate, it must have been creating an easy to use user interface without having access to any framework; building everything from scratch. On an embedded platform, which the Ultimate clearly is, you can't use standard frameworks like the ones commonly used in Java and C#, so you have to make one of your own.

Hmm, another obstacle was the development of a factory test system for the Ultimate-II+. That took quite some time. But then, I do think it saves me a lot of time.

Another one was the move to a web-shop system, rather than just taking orders and processing them manually.

What was/is your favorite game for the C64? Do you still find some time to play?

Giana Sisters... err.. no time to play!

I imagine that you do own a collection of stock C64s (i.e. all versions: from C64 "breadbin" with all the ASSY board revisions, to C64c, C64g and C128) for testing purpose. Are you a collectionist of retrocomputers as well, not only Commodore branded?

My wife would kill me, if I were actually collecting more. I only have working C64 mainboards, of which I use mainly just one in a C64C case. This has been the same machine as I used to test over 3000 ultimate's over the years. The power switch and cartridge port are a bit sad now. I do have a C128 and a C128D, but I never use them. I do have several floppy drives, too.

Can you even think about calculating how many hours you spent designing and working on the several versions of the 1541 Ultimate cartridge? What about the Ultimate64?

It is very difficult. As I said, many design activities take place as a background task. If I would count only the hours that I spend on the PC it might give a falsely low figure. What I can tell, tho, is that hardware designs, board layouts and such, usually don't take that much time. I think I created the U64 board design in about 3 weeks' time, but then of course only in the evenings and weekends. The schematics took a similar amount of time. Most time spent on technicalities goes into FPGA design, implementation and debug and firmware implementation.

From your question I sense that you focus a lot on the technical aspects, but I can tell you that the administrative tasks, including shipping orders and answering e-mails takes up most of my time, unfortunately.

Have you ever worked or are you planning to work on other projects involving the C64 or even different 8/16-bit machines?

Nope... :-)

How many people currently work at Gideon Lab on producing, testing and selling the two main products? Did you ever work in a team or simply get consulted with other electronics/software experts in order to achieve a particular result or to solve a bug?

Production is outsourced to a number of companies. (Production-) testing of the Ultimate-II+ is also performed in the factory. Production test for the U64 doesn't exist yet as of today, but that will be the next step in order to accelerate the process. When we talk about assembling the U2+ into plastic cases, that's often done by my wife, ... when she feels like it. She also plays an important role in packing orders. The other things are done by me; there are no employees at this point. Whether this can continue like this, is questionable. I think I do need external help for the quantity of U64's that are currently on order.

On the technical aspect, I sometimes talk with my colleagues about certain bugs, and of course I use the feedback and input from the community. There are some pretty smart guys out there that help me solve bugs sometimes. In order to achieve a particular result, I often apply patterns that I quietly pick up or learn from other projects.

Looking back to where you started it all, is there something that you regret about the PCB design or any other detail? Would you do something in a different way now if you could?

I mostly regret not taking the C64 FPGA code that I had made years before the U64 to a production level. I actually demoed a complete C64 in FPGA already back in 2011. I thought nobody would be interested in buying an FPGA-based C64 motherboard, since original C64 machines could be picked up for almost nothing, or else people would use an emulator anyway.

Regrets about other aspects: well, in retrospect many things could be regretted. But I think it is not fair to look at things like that, because as a person and as an engineer, you learn while you do it. Once you think things have to change, there is always the freedom to do so. I think that is one of the very cool things about having your own product. But I guess this principle applies to many things in life, doesn't it..?

I'm pretty sure that you worked very hard on both your projects during the last few years but also that you had so much fun doing it. What is the most funny/weird moment or story that you've been through while developing your products?

Oh, I absolutely had much fun doing it! Technically speaking, I have the most fun doing the FPGA code, second the hardware itself, and third the firmware.

I think one funny moment was the moment I realized how naive I can be. In the whole process of creating the 1541 Ultimate, I *never* thought of actually making a sellable product out of it. Or let's say, that was not my goal; it had always been pure hobby until then. It was actually a Swedish scener, TwoFlower, who happened to visit the Commodore Club in Maarssen just when I was giving the demo of a cartridge with an embedded floppy drive. He said I should have it produced, but I was hesitant and thought that it was not even feasible to do so. He asked me how many needed to be produced, and I stammered, "maybe 40 or 50?" He smiled and said: "Just do it… I'll make sure you'll sell all 40 of them in Sweden alone!" And that's how it all started!

Gideon, thank you very much for your time. This interesting interview ends here. Would you like to add anything, or say anything to our readers?

There is one important thing to mention.. I would like express a huge 'thank you' to the Commodore loving community. One of the most rewarding aspects of this project is the great feedback, the positive words I receive. In short: without you guys, I would never have been able to do all this. Thank you.

Interviewer: David La Monaca (aka Cercamon)

RetroMagazine Facebook Group - https://www.facebook.com/RetroMagazine-2005584959715273

Website - http://www.retromagazine.net

Interview made in May-June, 2018 © 2018 David La Monaca, Gideon Zweijtzer and RetroMagazine

Download the interview in PDF format (off-print from issue 8 of RetroMagazine) here: __ http://bit.ly/2NJXsEe

FAQ FOR THE ULTIMATE II AND ULTIMATE II PLUS CARTRIDGES

The **Ultimate** cartridge is a storage solution for your Commodore 64 home computer and will give you an excellent experience using this great machine. Some people even enjoy using the Ultimate cartridge with their Commodore 128 computer.

CONTENTS

15.1 Information and resources for the Ultimate devices

Please read Gideon's story about the Ultimate products. Read the specifications of the original Ultimate-II Plus cartridges. Read the announcement of the Ultimate-64 device and the update. Also read the Ultimate-64 project status.

You can buy Gideon's products in his web shop. You can also buy accessories.

15.2 Make sure you have the latest firmware installed

Pleaes make sure you have the latest firmware installed. This F.A.Q. only covers subjects relating to the latest firmware.

15.3 Where can I find the latest firmware?

You can find an overview the firmware releases for the Ultimate products on this page: firmware overview.If for any reason the ultimate64.com website is down, you can also download the firmware on github.

15.4 I am still running firmware 2.6k or lower on my 1541U-II/U2. How can I upgrade?

Upgrading from 2.6k to 3.x is a bit different from what you're used to. You will first need to upgrade to version 1541U-II Update 3.0 beta 7 using the update.bin in the root of your microSD card.

Once you've done this, you can download latest firmware and run the file that ends with .u2u to upgrade.

To revert to version 2.6k, use the revert.u2u that's included in the 3.0 beta 7 archive.

You can find the 3.0 beta 7 firmware on GitHub.

15.5 How can I use the emulated SID as second SID for 2SID tunes?

The question: link to facebook posting.

I'm looking for information on how to use the emulated SID of the U2+ together with a real one to play 2SID files, but so far haven't found enough to make it happen. I've fiddled with the settings and read the Sidplayer instructions but all I get is one (the C64 mainboard) Sid. Anyone know where I should look?

1. Find out what address the second SID is using for the 2SID you'd like to listen to. The ultimate sid player that's included shows the second sid address since firmware 3.2.

2. Go to the Ultimate menu and press F2 (shift + F1) to enter the configuration menu.

3. Select 'Audio Output settings'

4. Enable SID Left

5. Change the address at SID Left Base to the address you saw in the ultimate sid player. Probably something like $D420 or so

6. Go to 'SID Left Combined Waveforms' and choose the model the 2SID tune was composed for. Most likely 8580 since there are not many 6581 2SIDs.

7. Go to the Left Channel Output and Right Channel Output and select for both 'Left SID'.

8. Make sure to plug a mini-jack in the line-out mini-jack port and connect it to your powered/active speakers or an amplifier.

9. Run the 2SID file and have a listen ;-)

Btw, the address of the SID chip inside your c64 is $D400.

15.6 Does the U2 and U2+ support Easyflash Cartridge images?

Yes, both carts support EF. Since firmware 3.2 there is also support for saving to the EF cartridge image:

Technology Preview: EAPI Support for Easyflash, incl. but not limited to writing support. Please note that the changed crt file is NOT saved automatically. You can save it by "F5", "Save Easyflash crt" in the file browser. The file will be saved with the filelname "module.crt". Optionally rename the saved file to a more recognizable name.

Tip: turn off Ultimate Audio in the C64 and cartridge settings when using this feature.

15.7 Which USB2LAN adapters are supported by the 1541Utlimate 2?

Any USB2LAN adapter that uses an AX88772/-A/-B chip should work.

15.8 Can I control the Utlimate using a USB keyboard?

Yes you can!

You can connect a USB keyboard to the Ultimate cart and use that to control the Ultimate instead of using the buttons. Use the **scroll lock** key to enter the Ultimate menu and **ESC** will bring you back to the basic screen. You can use the keyboard to navigate the Ultimate menu, just like when you're using the remote interface via telnet, or using the C64 keyboard.

15.9 Can I play SID files from the High Voltage Sid Collection?

Yes, the **Ultimate** comes with a built-in SID Player called **The Ultimate C64 SID Player**.

15.10 How do I use the built-in SID The Ultimate C64 SID Player

As of firmware 3.2 The Ultimate C64 SID Player has keyboard support.

Use the following keys:

key	description
	fast forward
1 - 0	sub tune selection for tune 1 - 10
+	play next sub tune
–	play previous sub tune
run/stop	go back to Ultimate menu
space bar	pause / resume tune

Keyboard support only works for PSID tunes and for RSID tunes that don't run in a loop and when there is enough memory for the player. If the keyboard doesn't work for a particular tune, then press the cartridge button (default middle button) to go to the Ultimate menu for selecting another SID or sub tune.

15.11 Can I play SID tunes using the network interface?

There are several ways to play SID tunes using the ethernet interface:

- Use the remote interface using telnet;

- Use the Acid 64 Player Pro SID player by Wilfred Bos.

This SID player has the ability to play SID tunes over the netwerk on one or more Ultimate devices. It is built to run on the Microsoft Windows Operating System. When using Wine, Acid 64 Player Pro will also run on macOS and Linux. This for version 4.0 and higher it is unknown if it runs using Wine.

- Use Ultimate1541 Sid Remote 1.1 by L.A. Style of Genesis Project. This software runs on Microsoft Windows Operating system.

15.12 How do I use T64 files?

T64 is a file format, or rather a container, just like the D64 disk image. Unfortunately the U2 and U2+ do not recognise it as such. This means that you cannot open this container using the return key and then choose enter.

Use the right cursor key to enter the T64 container and then press return for the contextual menu to pop up and then choose run to run the c64 programme.

15.13 How do I create a create a directory

To create a directory on the file system browse to the location you would like to create a directory. Press F5 and choose the 'Create Directory' from the contextual menu.

15.14 How do I create a formatted disk image

To create a formatted d64 or g64 disk image on the file system browse to the location you would like to create a directory. Press F5 and choose the 'Create D64' or 'Create G64' from the contextual menu.

15.15 What does insert blank disk do?

Inserting a blank disk in either drive A or B will insert a blank and unformatted disk image in the drive. This means you need to format the disk before you can use it. Please take note that the disk does not exist on the filesytem (yet). If you saved files on the disk, then make sure you save the disk before turning off the computer.

15.16 How do I copy files using the Ultimate file browser?

The Ultinate device allows you to:

- copy files and directories accross the filesystem;
- copy files and directories accross the microSD card and USB thumb drives.
- copy files from the filesystem to disk images
- copy files form the disk images to the filesystem

Use these keys for copy operations:

- use space bar to select the file(s) / dir(s) you'd like to copy
- use cbm + c to copy
- use cbm + v to paste.

15.17 How do I rename files and directories on the file system?

To rename either a file or a directory simply select the file or directory and press enter. A contextual menu will pop-up, select 'rename' to rename the file.

15.18 Where are the manuals for the Ultimate II and Ultimate II + carts?

This FAQ is part of the current manual.

For both the Ultimate II and Ultimate II+ a quick start guide exists.

Another resource for some (old) documentation can be found here: http://rr.c64.org/wiki/1541_Ultimate# Documentation

15.19 Where are the manuals for the included emulated cartridge images?

You will need to google for that. Archive.org seems to have scans of several c64 cartridges.

The wiki rr.c64.org is also a great place to find manuals.

15.20 Flip/swap disk (mount the next disk without using the Ultimate menu)

Since firmware 3.0e there is this nice feature called: "seamless disk swap by pressing middle button at least 1 sec"

This allows you to mount the next disk when a game or a demo asks for it without entering the Ultimate menu.

It only works on disk images for which it is obviously that those disk images belong together E.g. "special game disk 1.d64" and "special game disk 2.d64" or "special game A.d64" and "special games B.d64" or "special game S1.d64" and "special game S2.d64". or "image 1.d64" and "image 2.d64", etc. etc. etc. It even recognises roman numbers.

15.21 How about the C128 and C128D(CR). Does the Ultimate cartridge work on these machines?

Yes, the Ultimate cartridges work on the C128, C128D(CR) but with some limitations. Bart was so kind to write down his findings and advice on how to get the most out of your C128 combined with the U2+: https://www.bartsplace.net/content/publications/1541ultimate128.shtml

15.22 Disabling the internal C128D drive

source: 1541ultimate.net forum

Most of the games and demos will only run from device #8. Even if you added a device number 8/9-Switch to your internal C128 drive, some games and demos do not like if there is a 2nd drive on the bus.

So, the good news: With S.T.F.U. you can disable your internal drive by software: https://csdb.dk/release/?id=160842)

1. Set your 1541-U drive emulation to device #9 or OFF

2. Start and run the file. (In my case I had to load it from disk - it does not seem to work if you DMA-start it from your 1541-U).

3. Select and deactivate your internal 1571.

4. Set your Drive A of your 1541-U to device #8 and use it as regular drive.

This did work on my C128D. It did not work on my SX64. However it might be of interest for C128D users.