



**User manual :**

# 1541-III

**for firmware version V08.02.xx or later**

**<http://jderogee.tripod.com>  
[jd1541@hotmail.com](mailto:jd1541@hotmail.com)**

**User manual :  
1541-III**

---

Date : Feb, 2008  
Version : 0.4  
Status : Final

## *Table of contents*

1	Introduction.....	6
1.1	Purpose.....	6
1.2	Scope.....	6
1.3	Definitions, acronyms and abbreviations.....	6
1.4	References.....	6
1.5	Overview.....	6
2	Unpacking the box.....	7
3	Disclaimer.....	8
4	Installation of the software.....	9
5	How to use the 1541-III.....	10
5.1	How to connect ?.....	12
5.2	Inserting the MMC/SD-card.....	13
5.3	How to set the device-ID?.....	13
5.4	Bootloader switch/jumper.....	13
5.5	How to show the files on the inserted MMC/SD-card ?.....	14
5.6	How to select a subdirectory and/or .D64 file ?.....	15
5.6.1	Selection using BASIC commands.....	15
5.6.2	Selection using buttons.....	16
5.7	My files don't show up ?.....	17
5.7.1	Fat-16.....	17
5.7.2	Subdirectories.....	17
5.7.3	File filter.....	17
6	Supported commands and their BASIC syntax.....	18
6.1	File selection without buttons or LOAD"\$"	19
6.2	Loading the directory of the current file.....	19
6.3	Loading the current (sub)directory of the card.....	19
6.4	File selection / loading dir. of a .D64 file using index.....	20
6.5	File selection / loading dir. of a .D64 file using filename.....	20
6.6	Loading a file from a .D64 file or from the root.....	21
6.7	Saving a file to the .D64 file.....	21
6.8	Formattting an existing .D64 file.....	21
6.9	Copying a file within a .D64 file.....	22
6.10	Removing a file within a .D64 file.....	22
6.11	Renaming a file within a .D64 file.....	22
6.12	Replacing a file within a .D64 file.....	22
7	Supported user commands and their BASIC syntax.....	23
7.1	"UI-" set drive into VIC-20 mode.....	23
7.2	"UI+" set drive into standard (C64) mode.....	24
7.3	"LCDCONTRAST" set the LCD's contrast.....	24
8	Software versions.....	25
8.1	Autodetection of board types.....	25

9 Trouble shooting ..... 26

10 Notes: ..... 27

## History

Date	Issue	Version	Description of changes
March 9, 2006	Draft	0.01	First creation
Oktober	Final	0.1	Added some real info...
Oktober	Final	0.2	Added ALL supported versions of the commands
November	Final	0.3	Added:set device ID using software set LCD contrast using software
Feb. 2008	Final	0.4	All sorts of changes, due to firmware improvements

## *1 Introduction.*

Thank you for your interest in the “1541-III”, a board designed for the 1541-III project, giving your CBM computer with IEC-port the possibility to access .D64 files stored on a MMC/SD-card.

This board has been designed by a Commodore fan for Commodore fans. The 1541-III project is an open source project and therefore development is never finished. So keep checking the website from which you’ve downloaded this manual for further details.

The 1541-III is a PIC18F2620 microcontroller driven, IEC-based, .D64 file, stored on an MMC/SD-card interpreting device that is partially compatible with the 1541 and 1541-III drive’s. Or in other words, the 1541-III is a standalone device that can directly access .D64 files as if it was an ordinary 5¼ disk. This all completely transparent for your CBM computer without the use of special cables connected to a PC and without special file handling tools on a PC.

### *1.1 Purpose.*

This document describes the functionality of the software and hardware.

### *1.2 Scope.*

This document is meant for explaining the functions, possibilities and ‘how to use’ of the “1541-III”. The reader(s) of this document and person(s) working with the “1541-III” is/are assumed to know :

- the basics of working with CBM computers and the related DOS commands,
- downloading and/or creation of .D64 files.

### *1.3 Definitions, acronyms and abbreviations.*

Not applicable

### *1.4 References.*

Not applicable

### *1.5 Overview*

Not applicable

## *2 Unpacking the box...*

Depending on how you obtained the 1541-III can be build by yourself or you've bought a kit or finished version. If it is put into a box, then probably the 1541-III will be the only thing that will come out of the box... along with a lot of bubblewrap plastic. But look further as the USB equiped 1541-III also has a USB poweradapter, that fit's into your CBM's cassetteport (saving you the trouble to connect it to a bulky power supply). During the reading of this manual you will discover that this project is nothing more then a simple piece of hardware and a huge chunk of software inside that nice little box with a very attractive display...

### *3 Disclaimer*

The “1541-III” is designed for use by persons as described in chapter 1.2. For safe and pleasant use follow the following rules :

- the “1541-III” is designed for indoor use only.
- the “1541-III” is designed to be used as an IEC device in combination with a CBM computer.
- do not connect your 1541-III to the USB-port of your PC **while** the 1541-III is still connected to the IEC-port of your CBM computer. This may cause interference and depending on the electrical grounding of your PC even damage to your CBM computer. Fortunately, there is no reason to set up this connection. The only time the 1541-III needs to be connected to the USB-port of your PC is when it needs to be upgraded with new firmware. Then the only connection required is the USB-port and the IEC-connector must be left unconnected. For more information regarding firmware upgrading, see the documents that accompany the firmware .hex file on the 1541-III’s website.



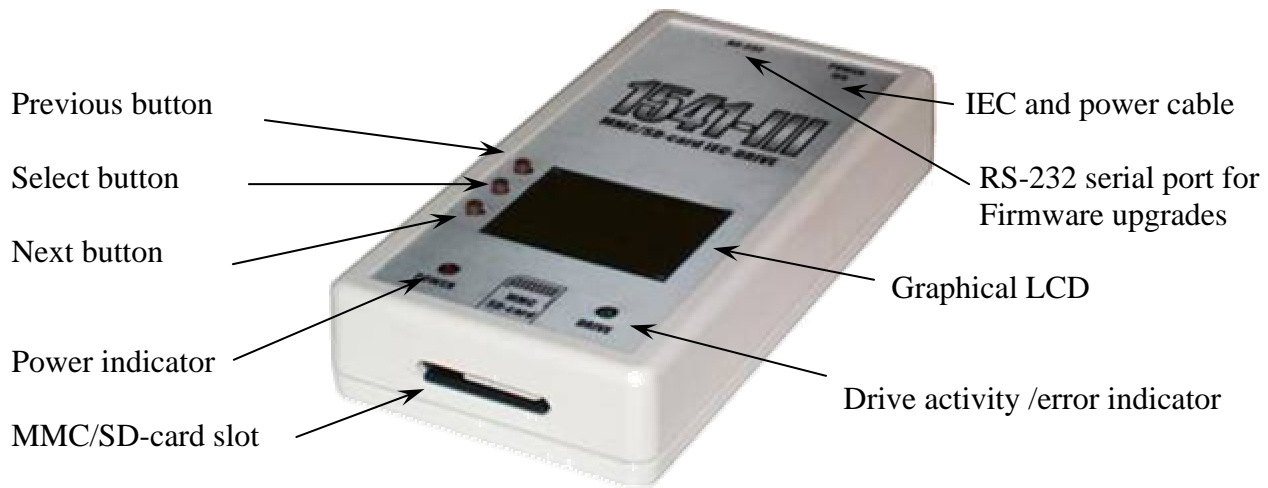
## *4 Installation of the software*

In order to make the 1541-III functional on your CBM computer no software needs to be installed. However for additional functionality it may be required to copy some extra files to your MMC/SD-card used in the 1541-III. On the 1541-III's website in the download section there is a tool named : Basicool 1541-III (.ZIP) This tool can execute some simple 1541-III settings.

For uploading new firmware the already programmed bootloader can be of help. Using this bootloader no extra programmer is required. Simply connect your board to your PC and install the special bootloader programming software and within a few minutes your 1541-III is up to date with the latest functionality (and bugfixes).

## 5 How to use the 1541-III

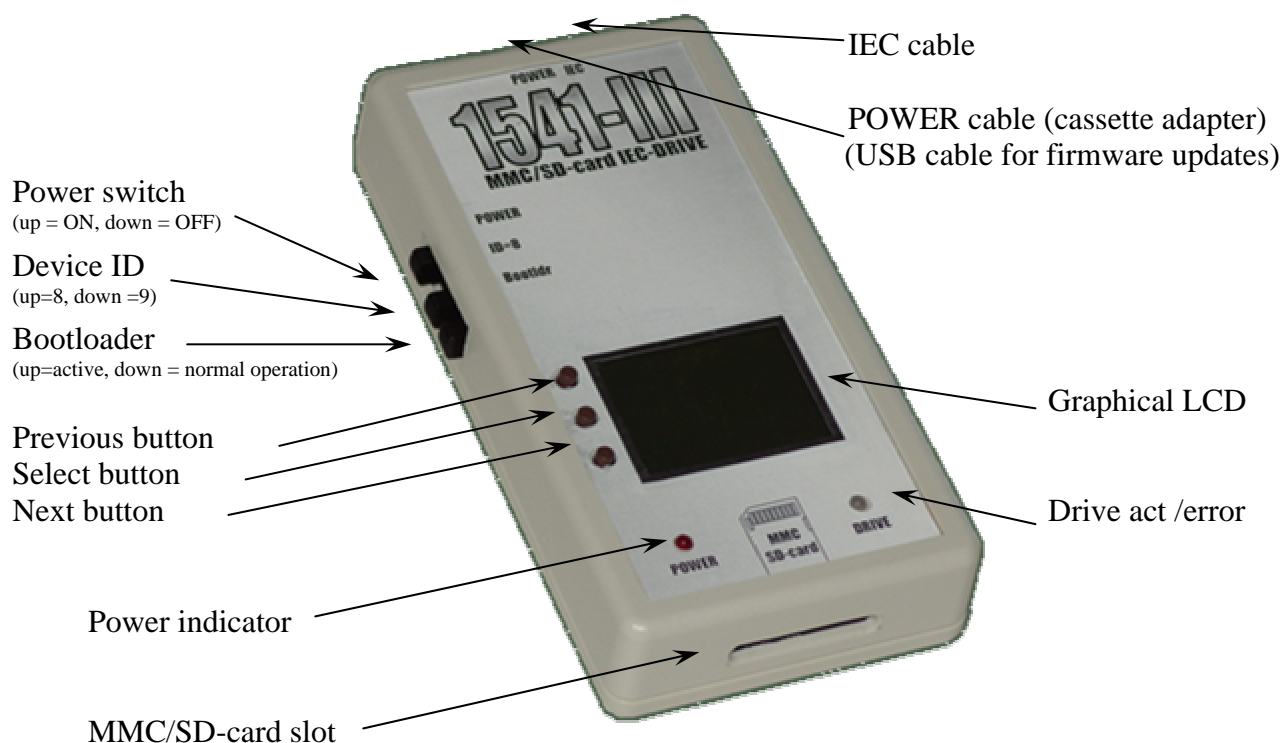
The 1541-III V1.0 (the version with the RS-232 serial port)



**Warning :**

Before connecting the IEC cable, make sure all related devices are powered off. Otherwise damage could be occur.

The 1541-III V1.1 (the USB version)



**Warning :**

Before connecting the IEC cable, make sure all related devices are powered off. Otherwise damage could be occur.

The second batch of 1541-III's all have 3 switches on the left hand side. The switch labelled POWER switches the 1541-III off. By switching it back on you force a hard reset, which is required during the firmware upgrading procedure. When the 1541-III is switched off, it should be disconnected from the IEC-bus otherwise it may interfere with the other IEC-devices.

## 5.1 How to connect ?

The 1541-III has 2 connectors, an IEC-connector and a power connector.

**IEC:** connects to the IEC-bus, better known as “serial port”, of your CBM computer. When other devices on the IEC-bus are required then place the 1541-III as the last device in the chain. This because the 1541-III has no pass through connector.

**Power:** connects to the power source.

### 1541-III V1.0 (RS-232 version)

The power connector is a pass through connector that draws it power directly from the CBM’s power supply.



### 1541-III V1.1 (USB version)

Because the 1541-III uses a USB connector for it’s power supply a small adapter is supplied. Connect this adapter to the USB connector of the 1541-III and then to the cassette port of your computer.



## 5.2 Inserting the MMC/SD-card

Gently insert the MMC/SD-card with the contacts facing up (see picture). Only use cards that are formatted according to the FAT-16 file format (blocksize of 512 bytes). The maximum card size that is supported by FAT-16 is 2Gbyte. The use of larger cards is not supported and will not function correctly.



## 5.3 How to set the device-ID?

The 1541-III v1.0 (RS-232 version) has a jumper inside.

**J2:** device            jumper placed device = 8,            jumper removed device = 9

The 1541-III v1.1 (USB version) has a switch to set the device ID to 8 or 9  
Simply move the switch in the up or down position to choose between device-ID 8 or 9.

The device ID can also be set by software. Do this by entering the following command sequence on your CBM computer:

```
OPEN15,8,15
PRINT#15,"M-W"CHR$(119)CHR$(0)CHR$(2)CHR$(<device#>+32)
CHR$(<device#>+64)
CLOSE15
```

Note: <device#> is the value of the new device number. This setting is lost when the 1541-III is powered off or reset.

## 5.4 Bootloader switch/jumper

V1.1 of the 1541-III has a switch to select the bootloader  
V1.0 of the 1541-III has a jumper inside.

**J3:** bootloader    jumper placed bootloader = active

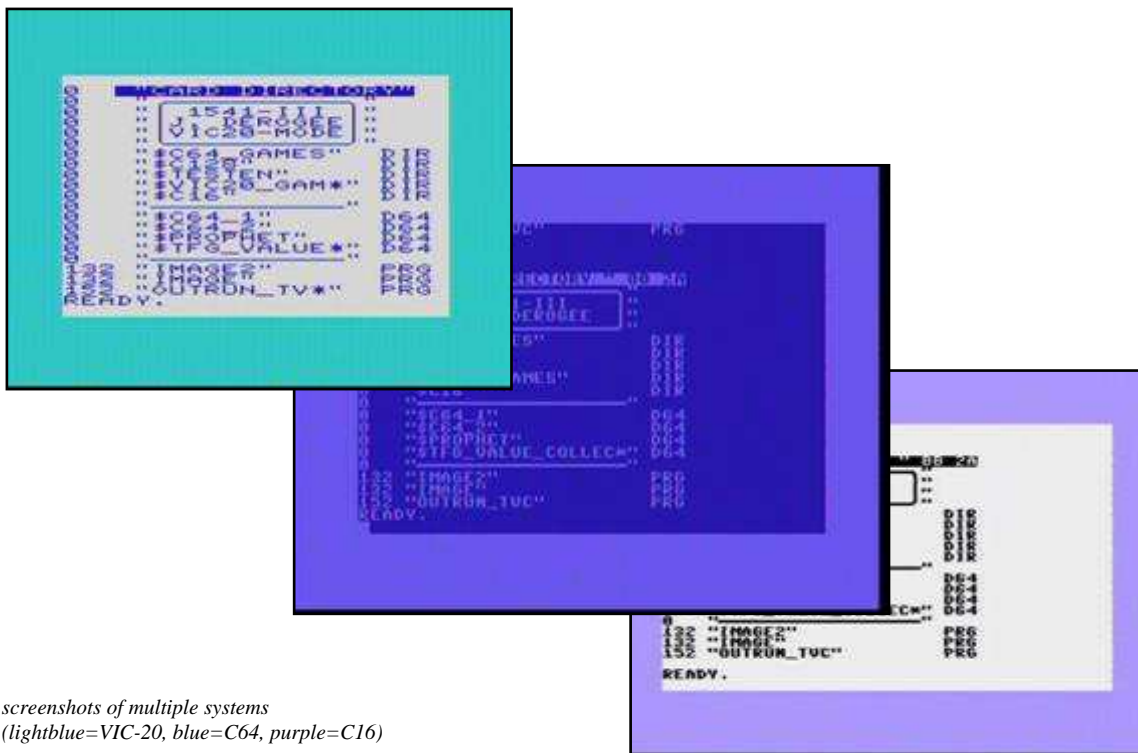
During normal operation the bootloader should not be active. When the bootloader is active it will (directly after power-on or reset) try to connect to the bootlader application. Using the serial or USB port during (max.) 5 seconds after powerup/reset the 1541-III will search for a valid connection. When no connection could be made the 1541-III will start automatically. For exact usage of the bootloader please refer to the document supplied with the firmware update.

## 5.5 How to show the files on the inserted MMC/SD-card ?

Normally the LOAD"\$" or LOAD"\$0" command loads the directory to the CBM's memory as if it was a program. And with the 1541-III it's just like that. With some small extra's. When no filename is given it refers to the currently selected file. But after a 1541-III reset or power-up there is no file selected and therefore it refers to the root-directory of the card. In that case it would be identical to the LOAD"\$CARD" command.

**LOAD"\$CARD",DEVICE**(where device is 8 or 9 depending on your settings)  
**LIST**

The screen will show a list of subdirectories, .D64 files and/or .PRG files as stored on the card. The number on the left shows the blocksize (only for files, not images or subdirectories). Your screen may look like the examples shown below :



screenshots of multiple systems  
(lightblue=VIC-20, blue=C64, purple=C16)

Filenames are shown according the LongFileName format, truncated to 14 characters. When a filename is longer, it ends with a \*-sign. This makes it possible to see that a filename is longer and in the meanwhile make it possible to load using normal basic loading commands, because the \* is a wildcard for the filesearching routines.

Subdirectories and images are preceded with the \$-sign, to indicate that these are to be loaded as a directory. Also this makes it easier to load, simply type LOAD at the beginning of the line showing the entry of interest and add the device number followed by a ':' afterwards.

## 5.6 How to select a subdirectory and/or .D64 file ?

### 5.6.1 Selection using BASIC commands

(if required) You can open a subdirectory.  
 Example the subdirectory with the name  
 "C64\_GAMES". Enter the directory load  
 command followed by the file name.

**LOAD"\$C64\_GAMES",DEVICE**

(the underscore (\_) is <CBM+@> on a C64 keyboard)

Now the subdirectory is openend. When  
 listed, you see that there are allways two  
 other entries that are not put there by the  
 user.



These are the entries "." and "..", these mean the following:

- . = the current subdirectory
- .. = the previous (also known as "parent" (sub)directory (one level back))

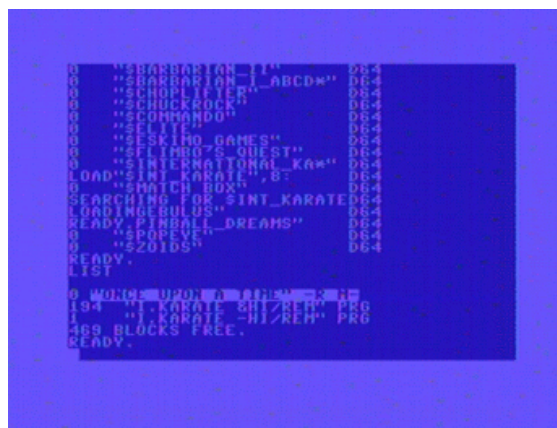
So when you want to return to the previous subdirectory:

**LOAD"\$..",DEVICE**

D64 files can be opened the exact same  
 way.For example the .D64 file with the  
 named "INT\_KARATE":

**LOAD"\$INT\_KARATE",DEVICE**

Now the .D64 file is openend and you  
 can use it as it was an ordinary 5¼ disk.  
 When you are inside an image (.D64) file  
 and want to return to it's directory use



**LOAD"\$CARD",DEVICE**

Now you will leave or de-select the D64 file and return to the card's location where the  
 previously selected .D64 file is stored.

### 5.6.2 Selection using buttons

An alternative to the BASIC commands would be using the 1541-III's buttons and its display. You can select another .D64 file by pressing the buttons previous- and next-file. When pressing these buttons the index on the top-line of the display changes accordingly. To indicate that the manual selection mode is active the file display turns into inverse mode. When the desired file has been found, select it by pressing the select button. Now the screen turns back into normal mode and the selected .D64 file can be accessed as if it was a real 5¼ disk.

During the manual file selection process the 1541-III cannot be accessed, this because manual selection mode ejects the currently selected disc. Unlike a real 1541 drive, it is not possible to select another .D64 file (eject the current 'disc') while the computer is still accessing the drive (i.e. during loading of a file).



In the same way as .D64 files, subdirectories can be selected. And when inside a subdirectory you'll notice the two entries "." and "..", these mean the following:

- . = the current subdirectory
- .. = the previous (sub)directory (one level back)

There is no use in using the "." entry, but it is shown because of historical reasons related to MSDOS. However the ".." entry is very usefull as it is needed to go back to the previous or parent (sub)directory. Selecting new subdirectories and/or deselecting them by using the ".." entry gives you the power to browse through the entire card's directory structure. But it is easier and more convenient to use the basic commands. It is best only use the buttons when the entry basic commands is not possible.



## 5.7 *My files don't show up ?*

### 5.7.1 *Fat-16*

The 1541-III only supports card that are formatted according the FAT-16 format (which means cards smaller then 2Gbyte) with a blocksize of 512bytes. So please make sure that this is correct (i.e. in Windows, request the properties of your card). If this is not correct all you can do is to re-format your card.

### 5.7.2 *Subdirectories*

The 1541-III does support subdirectories, simply because they do can be usefull and selecting a directory with only 3 buttons, does not have to be a problem. With cards getting bigger and bigger and with the limitation of only 512 files in the root, subdirectory support was unavoidable. But still... keep in mind that the best way is to manage your precious files is by doing so on your PC and to store only the most used files on your 1541-III's MMC/SD-card. Always keep a backup on your PC.

### 5.7.3 *File filter*

In order to prevent problems and to keep a better overview of the relevant files, the 1541-III has a build in filter that filters on the following criteria :

- files with the “.D64” extension and a filesize of exactly 174848, 175531, 196608, 197376 bytes
- files without extension or with the “.PRG” or “.T64” extension are not checked

## *6 Supported commands and their BASIC syntax*

The 1541-III supports the following commands, this chapter describes what these commands do and what BASIC syntax is required to put them to work.

The same function can be called in multiple ways, example

```
OPEN 15,<DEVICE>,15,"S:<FILENAME>"  
CLOSE 15
```

Should do exactly the same as the two line version

```
OPEN 15,<DEVICE>,15  
PRINT#15,"S:<FILENAME>"  
CLOSE 15
```

Practically these commands are equal to the user. But technically there are some small differences, these differences are also handled by the 1541-III but for the ease of typing this manual only the first option is given as an example. You are free to use whatever syntax you'd prefer.

## 6.1 File selection without buttons or **LOAD"\$"**

Syntax: **OPEN 1,<DEVICE>,1,"\$<FILENAME>":CLOSE1**

This command will select the .D64 file that is indicated by the filename

Example: **OPEN 1,8,1,"\$EXAMPLE":CLOSE1**

This selects the file named EXAMPLE.D64 file on the card.

## 6.2 Loading the directory of the current file

Syntax : **LOAD"\$",<DEVICE>**

Syntax : **LOAD"\$0",<DEVICE>**

This loads the directory of the currently selected .D64 file, or when no .D64 has been selected yet, the directory of the card itself.

Example: **LOAD"\$",8**

This loads the directory of the currently selected .D64 file, or when no .D64 has been selected yet, the directory of the card itself.

## 6.3 Loading the current (sub)directory of the card

Syntax : **LOAD"\$CARD",<DEVICE>**

This loads the (sub)directory of the card itself. When no subdirectory is selected, it shows the entries that are in the root. When inside a subdirectory is\t shows the entries that are in that subdirectory. Using this command is usefull when the user wants to exit the selected .D64 file (or image).

Example: **LOAD"\$CARD",8**

This loads the directory of the card/subdirectory

## 6.4 File selection / loading dir. of a .D64 file using index

Syntax : **LOAD"\$EXAMPLE",8**

This loads the directory of the <EXAMPLE> file on the card

Example: **LOAD"\$12",8**

This loads the directory of the 12<sup>th</sup> file on the card.

## 6.5 File selection / loading dir. of a .D64 file using filename

Syntax : **LOAD"\$<FILENAME>",8**

This loads the directory of the <filename> file on the card. The filename must be given according 8.3 format. But actually only the first 8 characters are used when searching for the file. This because there is a file check for D64 file, so a file must end with .D64 in order to be selected.

If the filename, as entered on your PC, is "PACMAN.D64", then the file is stored on the card as "PACMAN<space><space>D64". PACMAN padded with spaces to 8 characters and 3 for the extension D64. Hence the name of the format 8.3.

Example: **LOAD"\$PACMAN .D64",8**

This makes it possible to select the D64 name without knowing its indexnumber on the card.

Because the typing of the spaces can be forgotten easily, simply use the wildcard '\*'

Example: **LOAD"\$PAC\*",8**

## 6.6 Loading a file from a .D64 file or from the root

Syntax : **LOAD"<FILENAME>",<DEVICE>,1**

This command makes it possible to load a file from the .D64 file as well as loading a file from the root directory of the card. Loading a file from the card without selecting a .D64 file can be very convenient if you want to use a special tool to select a specific .D64 file.

Example: **LOAD"MY 1ST PROGRAM",8,1**

## 6.7 Saving a file to the .D64 file

Syntax : **SAVE"<FILENAME>",<DEVICE>,1**

This command makes it possible to save a file inside a .D64 file. It is not possible to save a file to the root directory of the card. Basically because this would make the 1541-III's FAT-routines much more complicated. Trying to save a file to the root directory of the card would generate an error.

If you have written a program that you want to save to a .D64 but you forgot to select a .D64 file, then use the 1541-III's buttons to select the appropriate .D64 file. **DO NOT** use the **LOAD"\$<INDEX>",<DEVICE>** command as it would load the directory of the .D64 file which as result overwrites the file you wanted to save.

Example: **SAVE"MY 1ST PROGRAM",8,1**

## 6.8 Formatting an existing .D64 file

**Note:** this command is designed for testing purposes and is not meant to be used by end-users. Empty .D64 files or formatting of .D64 files is best to be done using a dedicated .D64 editing tool (on a PC).

Syntax : **N:<DISKNAME>,<ID>**

Syntax : **N0:<DISKNAME>,<ID>**

Syntax : **NEW:<DISKNAME>,<ID>**

Syntax : **NEW0:<DISKNAME>,<ID>**

This will format the existing .D64 file to an empty .D64 file.

Example: **OPEN 15,8,15,"N:DISKNAME,01":CLOSE 15**

The diskname and ID values are not required. Make sure that when the ID is given it must be a 2 digit value otherwise the CBM will be confused when reading the directory.

## 6.9 Copying a file within a .D64 file

Syntax: **C:<COPYFILENAME>=<ORIGINALFILENAME>**  
 Syntax: **C0:<COPYFILENAME>=<ORIGINALFILENAME>**  
 Syntax: **COPY:<COPYFILENAME>=<ORIGINALFILENAME>**  
 Syntax: **COPY0:<COPYFILENAME>=<ORIGINALFILENAME>**  
 This command copies the specified file as found inside the .D64 file

Example: **OPEN 15,8,15,"C:WORKFILE.BAK=WORKFILE":CLOSE 15**  
 You can use the wild cards '?' and '\*' for the entry of the originalfilename only.

## 6.10 Removing a file within a .D64 file

Syntax: **S:<FILENAME>**  
 Syntax: **S0:<FILENAME>**  
 Syntax: **SCRATCH:<FILENAME>**  
 Syntax: **SCRATCH0:<FILENAME>**  
 This command removes a single or multiple files from the .D64 file

Example: **OPEN 15,8,15,"S:FILENAME":CLOSE 15**  
 You can use the wild cards '?' and '\*' to delete several files at once.

## 6.11 Renaming a file within a .D64 file

Syntax : **R:<NEWNAME>=<OLDNAME>**  
 Syntax : **R0:<NEWNAME>=<OLDNAME>**  
 Syntax : **RENAME:<NEWNAME>=<OLDNAME>**  
 Syntax : **RENAME0:<NEWNAME>=<OLDNAME>**  
 This command is to be used when renaming file within a .D64 file.

Example: **OPEN 15,8,15,"R:WORKFILE-BAK=WORKFILE":CLOSE 15**

## 6.12 Replacing a file within a .D64 file

Syntax : **SAVE"@0:<FILENAME>",<DEVICE>,1**  
 This command save the file to disk and overwrites the already existing file with the same name.

Example: **SAVE"@0:HIGHSORE",8,1**

For some reason some games like the usage of the syntax **SAVE"@S:<filename>",8,1** (i.e. commando, int karate, donkey kong, etc), although this is not stated in the 1541 and 1541-II manuals (making it an "illegal" command) it DOES work on the 1541 and 1541-II. Therefor in order to maintain compatibillity this "illegal" syntax must also be supported on the 1541-III.

## 7 Supported user commands and their BASIC syntax

### 7.1 "UI-" set drive into VIC-20 mode

Syntax : UI-

This command set the IEC bus protocol to a approx. 25% faster mode (data transfer over the bus, the actual measured time to transfer a program may be a bit slower due to the overhead caused by some fixed timings that do not change. The bus is faster but the 1541-III stays the same. So the effective speed gain is really noticeable on large files but may appear slower then a 25% speed gain.

The VIC-20 mode also has another feature. It makes the shown card directory smaller and easier to read on a small VIC-20 screen. Normally when you browse through a disks directory it does not fit onto your screen and every directory entry takes up two lines, making it difficult to read. The 1541-III in VIC-20 mode, limits the max. number of characters to a length that allways fit's onto your VIC-20's screen. However this is only for the card (sub)directories. For compatibillity reasons it has not been implemented in the display of the contents of images (.D64) itself.



a VIC-20 screen looks horrible in standard mode



a VIC-20 screen looks perfect in VIC-20 mode

Note: The VIC-20 mode setting is non-volatile because it is stored in the 1541-III's internal EEPROM. This means that you only need to set it once if you are a VIC-20 user. Because the VIC-20 mode uses a smaller screen layout and a higher IEC-bus speed it must be notified to the user that this mode is active otherwise the user may think that there might be something wrong. Therefor the text **VIC-20 MODE** is shown in the 3<sup>rd</sup> directory entry.

Example: **OPEN 15,8,15,"UI-":CLOSE 15**

This command will set the 1541-III into the VIC-20 mode this means that the device will operate on the IEC-bus with a IEC protocol with smaller delays, i.o.w. it goes faster. This only works reliably with a VIC-20, that's why its also called the VIC-20 mode.

## 7.2 “UI+“ set drive into standard (C64) mode

Syntax : **UI+**

This command set the IEC protocol to the “standard” mode as required for the C64. At default (when you received it) this mode does not need to be selected because it is already active.

Note: this setting is stored inside the 1541-III’s internal EEPROM it needs to be set in order to return from the VIC-20 mode.

Example: **OPEN 15,8,15,"UI+":CLOSE 15**

This command will set the 1541-III into the standard mode this means that the device will operate on the IEC-bus with the standard IEC protocol with normal delays, i.o.w. it functions on all CBM computers.

## 7.3 “LCDCONTRAST“ set the LCD’s contrast

Syntax : **LCDCONTRAST=<VALUE>**

This command set’s the contrast of the LCD value range is 1-254.

Example: **OPEN 15,8,15,"LCDCONTRAST=185":CLOSE 15**

The contrast setting is saved into internal EEPROM of the PIC microcontroller.



## 8 Software versions

### 8.1 Autodetection of board types

The 1541-III is also available in a 1541-III DTV version (as produced by Jason and Jussi), this version differs from the 1541-III. Normally this would mean there would be different firmware versions for the 1541-III and the 1541-III DTV versions. But because of the automatic board detection algorithm this is not the case. By detecting which board the software runs on, the software can independently decide if it runs on a 1541-III or a 1541-III DTV version. So it can show a different picture on the display and disable or enable some functions depending on the board it is running on.

This detection algorithm simply senses for the filter capacitor on port RA0, in a DTV this pin is open (no capacitance only the 5pF of the pin itself), the 1541-III has 100nF, so by setting the analog input to output and charge the capacitor then setting the output back to input and time how long it takes before the capacitor discharges we can determine pretty accurate what the size of the capacitor is. So in short the board detection is simply measuring the size of the capacitance on port RA0.

This means that there is room for multiple versions of the 1541-III. Although this method is not recommended design practice, it is in this case very useful for the constantly updated firmware, which now has to be only one version. This makes firmware revision control much easier.

## 9 Trouble shooting

Every system no matter how simple or complex, may sometimes not do what it is expected to. For these situations this chapter may be of use.

### **What is the exact terminal settings of the 1541-III?**

The correct terminal settings are 115200, No parity, No flow control.

### **When I connect the 1541-III to my PC's serial port and use a terminal program to test serial communication activity, nothing appears on my screen.**

Check if the correct cable is used. Some cables appear to be an "1 on 1 - cable" or "cross-cable" but have a different cabling inside. If you want to solder you own cable then the only connections required are RXD, TXD, GND. This means that you need to solder a cable with connections between:

- PIN-2 female (9-pole SUB-D) and PIN-2 male (9-pole SUB-D)
- PIN-3 female (9-pole SUB-D) and PIN-3 male (9-pole SUB-D)
- PIN-5 female (9-pole SUB-D) and PIN-5 male (9-pole SUB-D)

### **When I connect the 1541-III to my PC's serial port and use a terminal program to test serial communication activity, strange characters appear on my screen.**

Check if the selected baud-rate is correct, an incorrect baudrate may often lead to "scrambled" output. Attention, some terminals do not respond correctly when changing the baudrate to a new value. This may have been caused by the last received character, leaving the terminal in some "strange" wait-state. In these cases it is best to shut-down the terminal application on the computer and restart the application. This normally will reset the terminal while using the new settings.

Check if the used cable is not too long, the maximum useable cable length for an RS-232 is 50 feet (15 meter). This is under normal circumstances. When the cable is used in a industrial environment EMC (Electro Magnetic Compatibility) regulations are different and the cable may experience more interference. In this case a shorter cable should be used (maybe even in combination with additional filtering). A different path of the cable (leading around the interference source) may also be of help.

*10 Notes:*