
Software Data Interface Specification

for the

Multiple Classic Computer

Version 0.5 draft

Prepared by Dirk Dudenbostel

Arcade Retro Gaming

August, 30th 2010

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Intended Audience and Reading Suggestions	1
1.3 Project Scope.....	1
1.4 References	1
2. Overall Description	1
2.1 Product Perspective.....	1
2.2 Product Features	2
2.3 Folder structure for the MCC	2
2.4 System core Format	5
2.5 Supported Format for Picture and Sound Files.....	5
2.6 Support format for Game ROMs.....	5
2.7 Supported format for Documentation.....	5
2.8 Supported File format and endings	5
3. Retro Platform Simple Storage Format	5
3.1 Organization of Files.....	6
3.2 RP-INDEX.INI.....	7
3.3 RP-MEDIA.ID	8

Revision History

Name	Date	Reason For Changes	Version
Dirk Dudenbostel	Oct, 16 th 2009	Updated and Review	V0.1
Dirk Dudenbostel	Oct, 25 th 2009	Input from Cloanto, update of format	V0.2
Dirk Dudenbostel	Dec. 30 th 2009	Include .ARG system file description	V0.3
Dirk Dudenbostel	July. 27 th 2010	Update SD_Structure and .ini File	V0.4
Dirk Dudenbostel	Aug. 30 th 2010	Expand SD_Structure and .ini File	V0.5

1. Introduction

1.1 Purpose

This document describes the data structure, file formats, and folder structure for the Multiple Classic Computer (MCC) design.

1.2 Intended Audience and Reading Suggestions

This document is intended for cooperation partner from Arcade Retro Gaming. The purpose of this document is a standardized data structure, file formats and folder structure for the individual systems software released from Arcade Retro Gaming for the MCC.

1.3 Project Scope

MCC is a Multiple Classic Computer platform designed for real hardware based emulation of classic computers. The purpose of the MCC is to preserve the classic gaming experience inside the 21st century. Standardized interface should enable all software providers to support the MCC design. Future growth will enable different classic computer realizations based on the same fundamental interface.

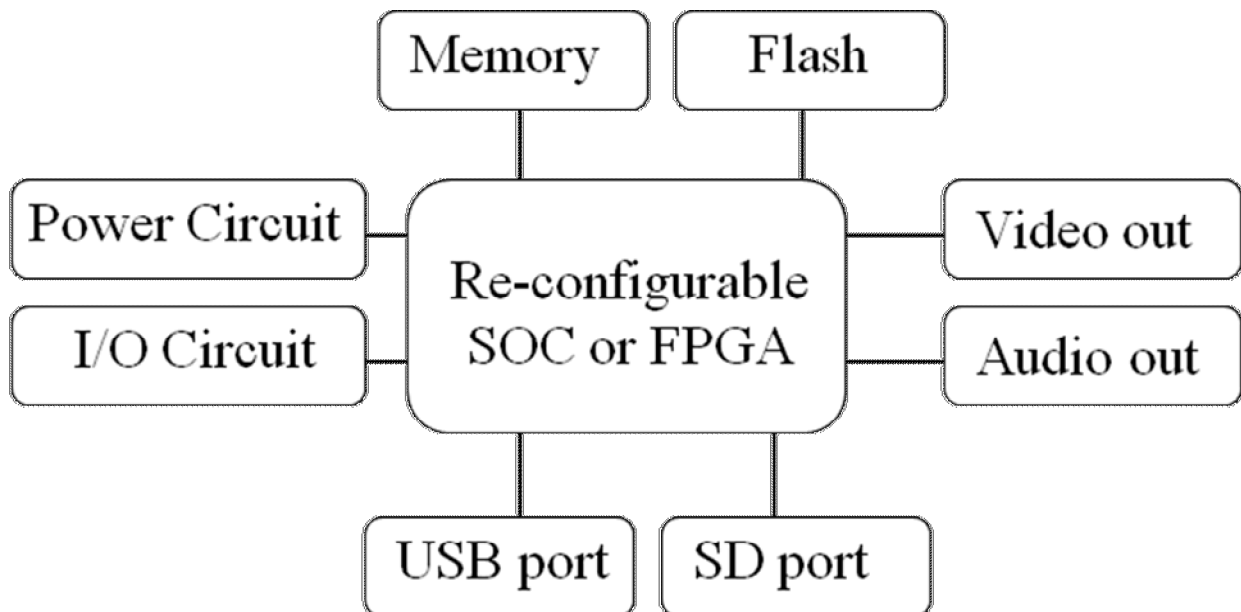
1.4 References

2. Overall Description

2.1 Product Perspective

The Multiple Classic Computer is a stand alone system which can be connected to every TV or Computer monitor. It support two classic DB-9 Joystick input ports for up to two individual players. Furthuremore it has interfaces for a mouse and a keyboard based on the PS-2 interface standard. An external USB Device port offers up additional connectivity options for more modern input devices. The support of USB Flash memory devices is not planed in the initial Software release. The main data storage is an external micro SD card. This SD card host all system relevant data including classic computer images, HMI menus, classic computer files, pictures, etc.

The following picture shows a block diagram of the hardware of the MCC.



Picture 1: Block diagram of the Multiple Classic Computer

2.2 Product Features

The Multiple Classic Computer platform enables different hardware emulation with a Field Programmable Gate Array (FPGA). The system has some internal Flash to store a boot loader application which loads the main classic computer design from the SD card. Furthermore an HMI enables the user to select the classic computer platform and the application in an easy way.

2.3 Folder structure for the MCC

This chapter describes the folder structure for the MCC:

SD-card:

- About(Optional)
- System
 - Ø Bootloader.arg
 - o Core
 - § C64
 - Ø C64PAL.arg
 - Ø C64NTSC.arg
 - § Amiga
 - Ø AmigaPAL.arg
 - Ø AmigaNTSC.arg

- § ATARI2600
 - Ø ATARI2600PAL.arg
 - Ø ATARI2600NTSC.arg
- § Classic_Computer_XY
 - ROM
 - § C64
 - Ø System_roms_C64
 - § Amiga
 - Ø System_roms_AMIGA
 - § Classic_Computer_XY
 - Skin
 - Picture
 - Sound
 - Font
- C64
 - Ø RP-INDEX.INI
 - Ø RP-MEDIA.ID
 - APPS
 - § Game1
 - Ø Gameraom1.d64
 - Ø Preview.bmp
 - § Game2
 - Ø Gameraom2.d64
 - Ø Preview.bmp
 - § ...Game_n
 - Ø Gameraom1.d64
 - Ø Preview.bmp
- AMIGA
 - Ø RP-INDEX.INI
 - Ø RP-MEDIA.ID
 - APPS
 - § Game1
 - Ø Gameraom1.adf
 - Ø Preview.bmp
 - § Game2
 - Ø Gameraom2.adf
 - Ø Preview.bmp
 - § ...Game_n
 - Ø Gameraom1.adf
 - Ø Preview.bmp
- ATARI2600
 - Ø RP-INDEX.INI

- Ø RP-MEDIA.ID
 - APPS
 - § Game1
 - Ø Gamerom1.prg
 - Ø Preview.bmp
 - § Game2
 - Ø Gamerom2.prg
 - Ø Preview.bmp
 - § ...Game_n
 - Ø Gamerom1.prg
 - Ø Preview.bmp
- USR_Data
 - Files
 - Pictures
 - Sound

The About folder contains a .txt file which describes the copyrights, link to Homepage and the version number of the Software / hardware.

The **SYSTEM** folder hosts the whole MCC software including everything for the boot menu:
The Boot loader should be placed into the System route directory.

The **CORE** subfolder contains a subfolder for each classic computer. This subfolder contains a binary file (.arg) which can be updated via the boot loader.

The latest classic computer cores can be found under www.mcc-home in the download section.

The **ROM** folder contains a subfolder for each classic computer. This subfolder contains all needed system ROMs for this computer.

The **SKIN** folder contains the HMI for the Boot menu

The **PICTURE** folder contains all needed pictures for the initial boot menu and navigation icons

The **SOUND** folder contains all classic sound for playback

The **FONT** folder contains all different fonts for the Boot menu

The **C64, Amiga, Atari2600, etc.** folders hosts all MCC games, demos and application software exported from C64 Forever or Amiga Forever:

Each root folder contains two files. One file called RP-Index.INI contains all programs, demos and application specification. The RP-Media.ID describes in detail the media and unique ID number. Please refer to the specific section talking about the RP-Index.INI and RP-Media.ID format and structure.

The **APPS** folder contains all Game roms and preview pictures in bmp format. Each game will be exported from the C64Forever or Amiga Forever Software in separate folders.

The **USR_DATA** folder hosts all user related data:

The **FILES** folder contains all user related program data (e.g. USR programs, Documents, Tables, etc.)

The **PICTURE** folder contains all user related pictures

The **SOUND** folder contains different sounds and music for playback

2.4 System core Format

Inside the SYSTEM/CORE folder the different .ARG Files are stored in specific sub folders.

2.5 Supported Format for Picture and Sound Files

The MCC will support pixel based non compressed image format. The preferred format would be .bmj. Resolution would be 160x120 pixels and the support color depth would be 16 or 256 with a 262144 color palette.

Sound files are supported should be as well uncompressed sampled audio in a .PCM format (standard sampling formats are supported).

2.6 Support format for Game ROMs

C-64: Support format would be .D64 and the .T64 format for the game rom images.

Amiga: Support for .adf files for the roms. It is as well possible to support encoded ROMs, i.e. ROM + rom.key

Today just the load from the sd-card is supported. In the future some support from a USB Flash drive might be possible.

2.7 Supported format for Documentation

Documentation format support would be standard .txt files.

2.8 Supported File format and endings

The MCC support the FAT 12/16/32 File system. The MCC won't support the long file names.

Please refer link for more details about the FAT file system and file name structure:

http://en.wikipedia.org/wiki/File_Allocation_Table

3. Retro Platform Simple Storage Format

The different application software for the C64 and Amiga can be exported via the Forever Software Series from Cloanto. The Multiple Classic Computer supports Version 2.1 of the RetroPlatform engine. Latest updates on the specification can be found under <http://www.retroplatform.com/kb/19-105>. Please be aware that the ownership of this specification belongs to Cloanto Italia SRL.

3.1 Organization of Files

The following is a sample set of files as might be found after exporting data to a medium:

RP-INDEX.INI

RP-MEDIA.ID

/APPS/

/APPS/GAME1/DISK1.ADF

/APPS/GAME1/DISK2.ADF

/APPS/GAME1/PREVIEW1.BMP

/APPS/GAME1/PREVIEW2.BMP

/SHARED/

/SHARED/ADF/

/SHARED/ADF/AWB134.ADF

/SHARED/HDF/

/SHARED/HDF/AWB310.HDF

/SHARED/ROM/

/SHARED/ROM/A130.ROM

/SHARED/ROM/A310.ROM

/SHARED/ROM/CDTV-EXT.ROM

/SHARED/ROM/ROM.KEY

/PRIVATE/

/PRIVATE/<PID>/

```
/PRIVATE/<PID>/MODEL1/
```

```
/PRIVATE/<PID>/MODEL1/CORE1.BIN
```

```
/PRIVATE/<PID>/MODEL1/CORE2.BIN
```

```
/PRIVATE/<PID>/MODEL1/FIRMWARE.BIN
```

```
/PRIVATE/<PID>/MODEL2/
```

```
/PRIVATE/<PID>/MODEL2/CORE1.BIN
```

```
/PRIVATE/<PID>/MODEL2/CORE2.BIN
```

```
/PRIVATE/<PID>/MODEL2/FIRMWARE.BIN
```

System files (ROMs, ROM.KEY, floppy disk images, hard disk images, etc.) are in the "SHARED" folder, rather than being duplicated with each application instance. Compared to an [Amiga Forever](#) or [C64 Forever](#) installation on Windows, where applications are grouped by type, all applications, including games and demo scene productions, are in a single "APPS" folder.

The "PRIVATE" directory is for third-party use. Multiple <PID> directories may co-exist on the same medium (e.g. if the same SD card is used on different devices).

Preview thumbnails are included in the game directory in BMP format, 160x120 and 320x240 square pixels (two files), true color.

3.2 RP-INDEX.INI

This file is an index of the exported system and application data content, referencing both the short (8.3) and the long names of the exported files. In this way, a single RP-INDEX.INI file may be used by devices with different file system capabilities. Additionally, the file provides configuration information and human-readable application details.

The RP-INDEX.INI file is in INI format. This format was chosen because it is simpler to parse than XML. An INI file consists of sections ("[Section]") and keys ("Key=..."). Applications other than RetroPlatform Player instances generally only need to read this file, not write it.

The current format uses only one section type, namely [Application]. Each [Application] section describes an emulated computer system and the related files, including any application-specific disks. Information is included to select and configure supported computer models, and to group applications by type (games, demos, barebone computer systems, etc.) and genre (game categories, etc.)

For example:

[Application]

Type=game ; types: game, demo, application, system, etc.

Genre=platform

Title="Asteroid Invader II"

Entity="Acme Games"

Year=1986

OID=1.2.3.4.5.6.7.8.9

Preview160="APPS\ASTEROID\PREVIEW1.BMP","APPS\Asteroid Invader II\Preview160.bmp"

Preview320="APPS\ASTEROID\PREVIEW2.BMP","APPS\Asteroid Invader II\Preview320.bmp"

System=a-500

Video=ntsc

Port=0,mouse ; Port numbers are 0/1 for Amiga, but 1/2 for C64

Port=1,joystick

AddExpansion=a-501

AddRAM=1048576,fast

AddDrive=1,dd ; Add floppy units 1 and 2 (0 is built-in on A500)

AddDrive=2,dd

ROM="SHARED\ROM\A130.ROM","SHARED\ROM\A130.ROM"

ROMKey="SHARED\ROM\ROM.KEY","SHARED\ROM\ROM.KEY"

Harddrive="SHARED\HDF\TEST.HDF","SHARED\HDF\Test Long Name.HDF",1

Floppy="APPS\ASTEROID\DISK1.ADF","APPS\Asteroid Invader II\Disk 1.adf",1

Floppy="APPS\ASTEROID\DISK2.ADF","APPS\Asteroid Invader II\Disk 2.adf",1Notes:

References to file names always list the short name followed by the long name

The long names may include ISO 8859-1 characters that are not part of the 7-bit ASCII set (e.g. "APPS\CD³²\preview160.bmp")

Port settings and additional peripherals are always relative to the canonical system configuration (which for example may already include a floppy drive)

Preview images in both 160x120 and 320x240 format are provided (if in doubt, consider that 160x120 may be smaller but better quality, as 320x240 sometimes is the result of up-sampling)

3.3 RP-MEDIA.ID

This optional, short, fixed-width text file serves to uniquely identify a medium (e.g. an SD card) and the devices to which the medium was exposed (and therefore may need to work with).

Sample RP-MEDIA.ID file:

```
15C3FA0D-D4D2-11DE-9DEF-0050C2490000
```

```
MAXIMIGA-2____-0123456789AB-1____-1____
```

CLOANTO_-RP9X-0123456789AB-2009-1____Recognizing a medium (GUID in line 1 of RP-MEDIA.ID file) is helpful for synchronization purposes. (For example, Windows Media Player uses a file named SMPInfo.xml, and Windows Backup uses MediaID.bin for similar purposes.)

Having additional device information (after line 1, one additional line per device) can help RetroPlatform Player understand what application types and data formats are most suitable for export. In line 2 of the above example, the medium was tagged by device "2" (e.g. it could refer to the second product) of manufacturer "MAXIMIGA", having a device ID of "0123456789AB" (which could be a MAC address), and a version of "1.1". Underscore characters may be used for padding at the end of a field, and are ignored when parsing. Hyphen characters are inherited by the GUID syntax, and are similarly used to visually separate fields in the other lines, at fixed positions as in the example.

Devices are encouraged, but not required to use the RP-MEDIA.ID file. If the file does not exist (or if it is empty), RetroPlatform Player will create it. A typical file used on only one device is expected to only consist of two lines of text (one line to identify the medium, one line to identify the device). However, the format is designed to support cross-device portability and compatibility, so multiple devices can add one line each.

Because RP-MEDIA.ID is the most likely file that even the simplest devices may want to not only read, but also create (e.g. when formatting the medium), it is structured with a priority on easy seek and edit. More in detail:

All lines consist of 36 characters followed by a CR+LF (0x13+0x10) pair

The only valid characters are: uppercase letters A-Z, numbers 0-9, hyphen and underscore characters
Hyphen characters (0x2D) are used for visual separation at fixed positions, which are skipped when reading data

Underscore characters (0x5F) at the end of a group are only used for padding, and are ignored when reading data

For the first line:

This line is required

The line contains the 16-byte unique medium identifier GUID in uppercase hexadecimal notation (data is grouped as 8+4+4+4+12 characters, with a "-" separator between each group)

The GUID may, but does not need to, strictly conform to a specific GUID version guideline. Any sequence of random or pseudorandom data is OK.

For the other lines:

Each line after the first line is optional

Each line is device-specific (only one line per device)

The hex data is grouped as 8+4+12+4+4 characters, with a "-" separator between each group

The fields are: <PID> (8 characters), product ID (4 characters, decided by owner of <PID>), unique product-specific serial number or other instance (12 characters, e.g. MAC address), product-specific (e.g. firmware) major version (4 characters) and minor version (4 characters)

Once a device recognizes a line as belonging to that device (by combining the <PID>, the product ID and the product-specific serial number), it may edit that line (e.g. to update the version fields)

If a device does not find a line that references that device, it may append a line to the end of the file

If a device needs to delete a line, it may fill it with underscore characters (if moving all data is too complex)

A line consisting of 36 underscore characters can either be ignored or be considered available for reuse by any applications that needs a line

Devices that support RP-MEDIA.ID are encouraged to look for the file and create it if necessary. If the file is present, devices should scan the file, see if there already is an entry for that device, and add a line if not. If the device is listed, but the version information is incorrect, the line should be updated.