

## Audio file management in the archive

The MICMON2 firmware on-board MPT2/3/4 and MA45/46/47 cards includes file search functions in the USB drive previously formatted with Microsoft FAT16 or FAT32 and loaded with audio files using a PC. This operating firmware imposes the following restrictions on the folders and files in the USB drive:

- \* maximum 54 folders can be placed in the Root Directory
- \* maximum 999 files can be placed in the Root Directory
- \* maximum 99 subfolders can be placed in one folder
- \* maximum 99 files can be placed in one folder
- \* there are some restrictions on the names of folders and files which are described in the MICMON2 manual
- \* recommendation only – audio files should be placed either in the Root Directory or in folders on the last (highest) level.

The file search function depends on how many digits were entered for the searched file:

**1 to 3 digits entered** – the search is not for the file name but for the position of that file (which is determined by the order of writing files) in the selected directory.

**4 to 10 digits entered** – the search is for the file name.

In the Root Directory, up to 999 files can be searched for position using one (1-9), two (01-99) or three (001-999) digit numbers. Folders in the archive can store only up to 99 files which are searched for position using one (1-9) or two (01-99) digit numbers. When searching for file's position with 1-3 digits, the file's name does not matter. The files in the Root Directory can be searched for file names using 4 to 10 digits. The search for the file name with 4-10 digits places special requirements on the file name format as explained below.

In a PC-based environment, which features a full alphanumeric keyboard, the file names normally include letters and numbers. Searching for files named that way using a decimal keypad would be cumbersome and time consuming. Therefore, the file management system used in this archive has been designed to search for file names that include decimal digits only except for the first character. The first character in the file name can be a digit 1 to 6 or one of the following six letters: R, M, B, P, L and U (lower or upper case). MICMON2 firmware automatically substitutes the above letters with the following digits: R – 1, M – 2, B – 3, L – 4, P – 5, U – 6 (default values). These letters are allowed for user convenience only and could be used in master audio files on the PC for better differentiation between various categories of sounds. However, for the archive's search function, the file name R367 is the same as file name 1367.

These six letters define possible categories of audio files in the archive. The user may, for example, adopt the following six categories:

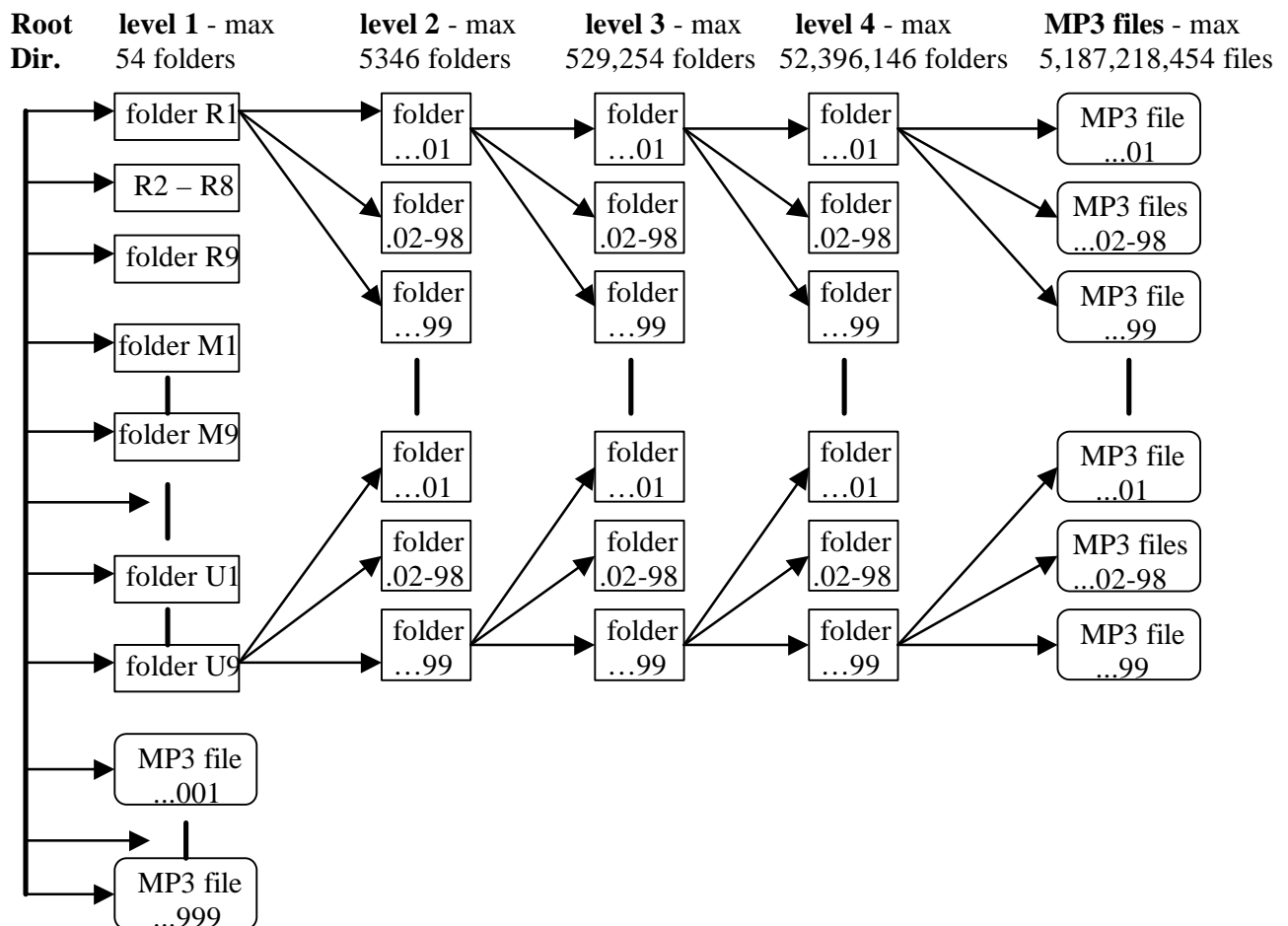
- R – Radio podcasts and recordings
- M – Music, classical
- B – (audio)Books
- L – Lectures and similar
- P – Popular music (pop)
- U – Unclassified

<b>1</b> (R)	<b>2</b> (M)	<b>3</b> (B)
<b>4</b> (L)	<b>5</b> (P)	<b>6</b> (U)
<b>7</b>	<b>8</b>	<b>9</b>
<b>*</b>	<b>0</b>	<b>#</b>

The names of folders created within the Root Directory must be exactly two character long like R1 (or 11), R2 (or 12), M1 (or 21), B1 (or 31), etc. The second character in the folder (digits 1 to 9, 0 is reserved) may be used to define a sub-category within one of the above categories. For example, folder R1 may be used for podcasts from one radio station, folder R2 from another station, etc. The same subdivision can be applied to the M category and so on. The remaining

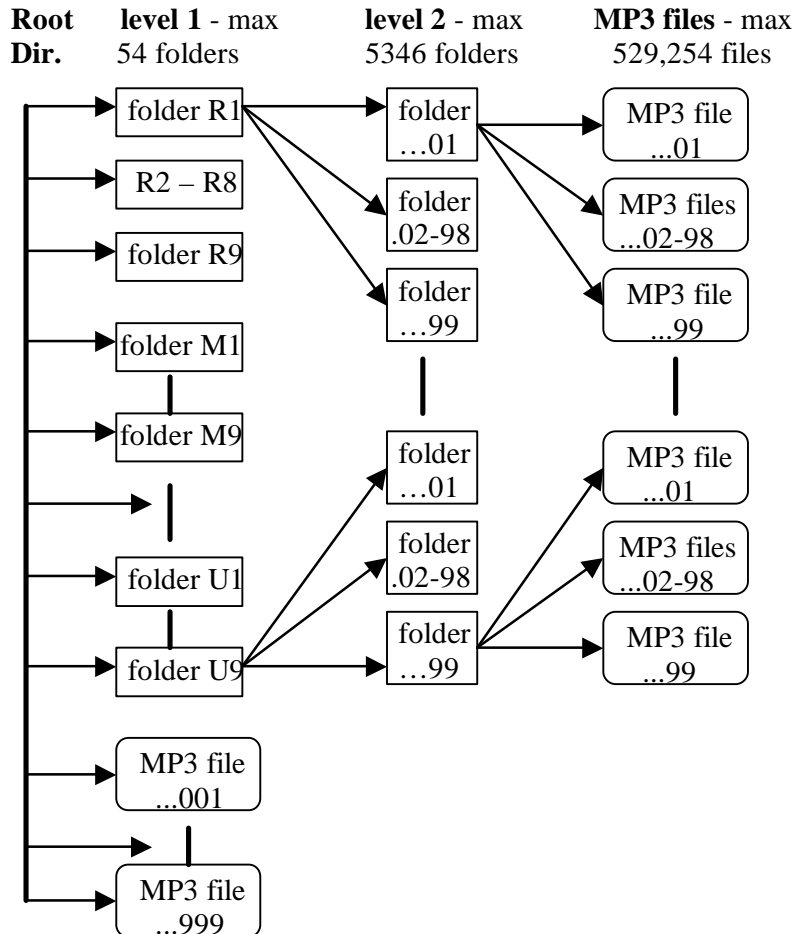
digits can be allocated to further sub-categories or simply used for numbering files within that category. The number of digits used in the filename is not fixed and can vary between categories. When searching for the file name, the Root Directory is searched first and, if the file is not found there, the folders and subfolders are searched next. The directory/folder structure recognized by MICMON2 could be up to four levels deep.

#### 4-level directory structure



The practical limit on the number of files stored in currently available USB hard drives with FAT32 is of the order of several millions. For example, storing 5 million files on a 1TB drive would allocate slightly less than 200KB (i.e., six 32KB clusters) per file taking into account memory occupied by FAT32 itself. A 3-level folder/directory structure would be then sufficient to accommodate many millions of files. However, a 4-level directory structure is also supported in order to make it easier for dated files to be archived and retrieved. This is relevant in applications where up to 99 files/day might be required to be archived, for example, radio stations podcasts, law court proceedings, lectures or various security recordings. For such dated files, the directory structure would look as follows. Level 1 would contain main folders, e.g., R1, R2, ... L1, etc. Level 2 would split the collection in folders on level 1 into years, for example, 31 folders would cover years from 1990 till 2020. Level 3 would split the year into months (12 folders) and level 4 would contain 30 or 31 folders corresponding to days for each month. The actual MP3 files are written in the day folders only. Such an arrangement allows for an easy maintenance of files.

## 2-level directory structure



Please note that different categories (R, M, B, ... U) can adopt different levels of directory structure. For example, R1 and R2 can be four-level, M1-M9 one-level, B1-B8 two-level and B9 one-level deep.

### Example with audiobooks

An audiobook collection of max 891 titles can be implemented with folders B1-B9 in a 2-level directory structure while leaving room to add other categories. Folders on level 2 (9x99=891 in total) hold one audiobook each. Normally, an audiobook consists of separate audio files (segments) which usually correspond to chapters. Each folder on level 2 would then contain several MP3 files being the audiobook segments.

### Example with lectures

If a total number of lectures does not exceed 99, they all could be placed in folder L1. The lectures' MP3 audio files should then be written directly into folder L1.

### **Example with radio podcasts**

If the archive is to store several podcasts a day from one station, it might be convenient to include the date in the filename. Using all ten digits available in the filename, it might have the following format:

R1080925xy where folder R1 is allocated to that station only, 080925 is the date (2008, Sept 25) and xy define each podcast on that day. The last two digits xy can be further divided into subcategories. For example, if on Sept 25, 2008 Australia's ABC Radio National (assigned to R1) had three podcasts available (science, health and religion), they could be named R108092511, R108092521 and R108092531.

### **Example of a small mixed archive with up to 891 files**

This collection is without folders and all the files are placed in the Root directory only.

#### **Searching for the file's name in the Root Directory using 4-digit numbers**

The files are searched for file names which, for example, could include the following 4-digit names:

R101 to R199 – podcasts from radio station 1 (searched for as 1101-1199)

R201 to R299 – podcasts from radio station 2

R301 to R399 – miscellaneous radio podcasts

M101 to M199 – classical music pieces

B101 to B199 – best suited for single file audiobooks or for only a few segmented audiobooks, a large number of segmented audiobooks would require folders as described in the Example with audiobooks

P101 to P199 – songs

P201 to P299 – songs

P301 to P399 – songs

U101 to U199 – miscellaneous audio files

Remembering about the substitution of the first letter in the file name, the song P248 should be searched for as having the file name 5248.

Please note that the search for the file name requires the file name to be from 4 to 10 characters long.

#### **Searching for the file's position in the Root Directory using 3-digit numbers**

Instead of naming the files as shown above (R101, R102 ...), they could be named with the first three characters being digits 001 to 999 and the remaining characters being any letters or digits, e.g., 258Mozartxxx. A collection of files in such a format could then be copied from the PC folder to the USB drive's Root Directory in one operation which would write the files in the sequence dictated by the first 3 digits in the file name. Such files could not be searched later on by their file names but only by their position. For example, the 258Mozartxxx file could be found by searching for the file position 258 in the Root Directory. Any changes within such file collection (deletion, replacing files, etc) would need to be done in the PC folder that contains a master copy of these files. The USB drive's Root Directory would be cleared and all the files copied again from that PC folder.

### **Example of mixing file name formats**

Files with decimal file names, formatted accordingly, could be placed in folders only and files with any names (alphanumeric or decimal) in the Root Directory only. The files in the folders would be searched for file names and the files in the Root Directory for position. In this arrangement, the Root Directory would be used as a kind of scratch pad for files that are often deleted and replaced with new ones.

## **Indexing the archive**

This embedded audio archive has been designed for use without display. It is possible to implement voice prompt menus and submenus to narrow down the choice of files but that approach would be best suited to specific applications. There are obviously some drawbacks of not having a display but in many applications a display would be of little or no use. For example, when used in a car, it is much safer and quicker to operate the archive with one hand only at the keypad, without taking the eyes off the road, than trying to select tracks on the MP3 player's tiny display. Similarly, books could be read on the screen but a hard copy is still more attractive. A major advantage of a hard copy versus menu-driven display is the ease and speed in locating the desired file. For this reason, a completed audio archive, either personal or commercial, should be complemented by one or more printed booklets listing file names with short descriptions. During use, the booklets could be hand annotated by users with comments or ratings of various recordings. If the archive's master audio files on the PC are named with the first character of the file name being R, M, B, ... it would be better to use only decimal digits in the file names in the archive's index booklet.

For example, a dated radio podcast (R108092511) should be listed in the index as 1108092511 – ABC science show .....