BigFAT

Large file support

for the FAT file system

Functional Specification

Document: RM02002
Revision: 4
Date: Aug 11, 2022

SEGGER Microcontroller GmbH

www.segger.com

Table of Contents

1 About this document

This specification proposes an extension to the FAT file system that allows an application to manage files larger than 4 GB in size.

2 Motivation

FAT is a popular file system that is widely used on resource-constrained devices thanks to the simplicity of its design and to the universal support provided by all modern desktop and mobile operating systems. Although it was designed more than 40 year ago, FAT is still suitable for most embedded applications. Nevertheless, it also shows some shortcomings, such as a limited file size.

This limitation becomes apparent for example in the usage of databases in embedded applications. With ever-increasing CPU speeds and memory capacities, the embedded applications nowadays normally use lightweight database engines to manage the data they have to process. A database can make the processing of the data easier for the embedded application, for example by taking over the tasks of storing, retrieving and manipulating of the data. Typically, a database engine uses a single file to store all the data which, depending on the amount of data that has to be stored, can grow to a few gigabytes in size. It is not uncommon for such a file to grow larger than 4 GB, but the file size limit imposed by the FAT file system is $2^{32} - 1$ bytes, and therefore makes it unsuitable for such applications. This limitation requires to either switch to another file system that supports files larger than 4 GB or to choose another database engine that is able to split the data across files smaller than 4 GB. Both solutions come with various risks and can considerably increase the cost of development.

In contrast to this, BigFAT provides a better solution by allowing the embedded application to create files larger than 4 GB, while still remaining compatible to the FAT specification. This means that an application that does not support BigFAT will still be able to correctly access the files created by an application using BigFAT. To use BigFAT in an application, a file system is needed such as SEGGER emFile that comes with support for it.

3 How it works

BigFAT is built on top of the FAT file system. It splits the data of a file that grows larger than 4 GB into files smaller than 4 GB. Consequently, these files can be stored as regular files on FAT file system. The file system operations are redirected to the appropriate file, depending on the file position that is accessed by the application.

Files with a size smaller than the maximum file size (4 GB – 128 KB) are handled like regular FAT files. In this case, the file access works as in any standard FAT implementation. As soon as the file grows above the limit, the file becomes extended. An extended file consists of one base file and one or more continuation files. The continuation files are stored in the same directory as the base file, with the name derived from the name of the base file plus an additional file extension. The name of the base file remains unchanged. The file extension of a continuation file has the format `.<Index>.BigFAT`, where the `Index` is a three digit decimal number. The index of the first continuation file is set to one and it increases by one with each additional continuation file. BigFAT requires support for long file names on the underlying FAT file system in order to be able to apply the additional file extension of continuation files.

When the base file grows larger than maximum file size, the first continuation file is created. This procedure is required in order to guarantee that the first byte in the continuation file is aligned to a logical sector boundary. This helps increase the performance of the file system accessing the file data. The same procedure is applied when the first continuation file exceeds the maximum file size, with the second continuation file being created and so on.

4 Examples

For example a 6 GB `Test1.txt` file is split into one base file and one continuation file. The size of the base file is set to 4 GB – 128 KB and size of the first (and also last) continuation is set to 2 GB + 128 KB. This file appears on a system without BigFAT support as two different files:

File name	File size
`Test1.txt`	4 GB – 128 KB
`Test1.txt.001.BigFAT`	2 GB + 128 KB

In contrast, on a system that supports BigFAT the two files are seen as a single file of 6 GB in size and have the name of the base file:

File name	File size
`Test1.txt`	6 GB

As another example, a 10 GB `Test2.txt` file is split like this into one base file and two continuation files:

File name	File size
`Test2.txt`	4 GB – 128 KB
`Test2.txt.001.BigFAT`	4 GB – 128 KB
`Test2.txt.002.BigFAT`	2 GB + 256 KB

5 Glossary

Term	Description
Base file	Regular FAT file that stores the first 4 GB - 128 KB of an extended file.
Continuation file	Regular FAT file that stores the data at off-sets larger 4 GB - 128 KB of an extended file.
Extended file	A file with a size larger than or equal to 4 GB. Each extended file is stored as one base file and one or more continuation files.
FAT (File Allocation Table)	Widely-used type of file system developed by Microsoft Corporation.