مرکز آموزش نیـــرا سیستم
nirasystem.com

FatFs

Ali Mirghasemi

Introduction

- FatFs is a generic FAT/exFAT file system library designed for embedded systems.
- Developed by ChaN, it provides a lightweight and portable solution to implement FAT12, FAT16, FAT32, and exFAT file systems on small microcontrollers with limited resources.
- FatFs supports various storage media, including SD cards, USB flash drives, and hard disks.

Applications

- **Data Logging**
  - Storing sensor data, logs, and other information on SD cards or USB drives.
- **Multimedia**
  - Managing audio, video, and image files on embedded devices like MP3 players and digital cameras.
- **Firmware Updates**
  - Facilitating firmware upgrade processes via removable storage media.
- **Industrial Automation**
  - Recording operational data and parameters in industrial control systems.
- **Consumer Electronics**
  - Enabling file management in devices like smart home appliances and wearable gadgets.

Properties

- **Compatibility**
  - Supports FAT12, FAT16, FAT32, and exFAT file systems.
- **Portability**
  - Designed to be easily ported to various microcontrollers and operating systems.
- **Efficiency**
  - Optimized for resource-constrained environments, with low memory and CPU usage.
- **Reliability**
  - Implements robust error handling and recovery mechanisms.
- **Flexibility**
  - Configurable features to balance between functionality and resource usage.

Configurations

- **Volume Management**
  - Supports multiple volumes and dynamic volume mounting/unmounting.
- **Sector Size**
  - Configurable sector size to match the underlying storage media.
- **Code Page**
  - Supports different code pages for character encoding.
- **Long File Name (LFN) Support**:
  - Optional support for long file names, increasing compatibility with modern file systems.
- **Synchronization**
  - Option to enable thread-safe operations for use in multi-threaded environments.
- **Timestamp**
  - Configurable timestamp settings for file creation, modification, and access times.

Library Structure

- **ff.c**
  - Core source file implementing the file system functions and API.
- **ff.h**
  - Header file defining the public API and data structures used by the library.
- **ffconf.h**
  - Configuration file where developers set various compilation options and parameters.
- **diskio.c**
  - Disk I/O interface layer providing functions for low-level media access.
- **diskio.h**
  - Header file defining the disk I/O interface.
- **Option**
  - Directory containing optional extensions and additional utilities for specific use cases.

APIs

- **f_mount**: Mount a file system.
- **f_open**: Open a file.
- **f_read**: Read data from a file.
- **f_write**: Wr
- **f_close**: Close a file.ite data to a file.
- **f_lseek**: Move the file read/write pointer.
- **f_stat**: Get file status.
- **f_unlink**: Delete a file.
- **f_mkdir**: Create a directory.
- **f_opendir**: Open a directory.
- **f_readdir**: Read a directory entry.