j1FAT32 - FAT32 Library Functions

v1.0

Features

- Object oriented library
- Read-Only FAT32 functions
- LongFileName (LFN) optional
- Device independent object linkage

General Description

An object oriented device independent FAT32 library. Based on the control object the functions will handle a read-only FAT32 device with optional LongFileName capability. Multiple control objects can be created and the shared library code will work independently with each object. Control objects can be local in scope or global storage areas for shared devices.

Legal & Disclaimers (Fine Print)

When to Use the j1FAT32 Component

You can use the j1FAT32 component any time you want a read-only FAT32 compatible library. The attached device must provide a readSector() interface to function. You can include it in another component and it will expose the functions with a defined library prefix ($j1LIB_NAME) instead of the $INSTANCE_NAME parameter. The default $j1LIB_NAME is j1FAT32.

Dependencies

j1FAT32 requires a j1Types component included in the top design of the project. A block oriented handler component will also be required to function.

Input/Output Connections

As a function library the component has no input/output connections.

Schematic Macro Information

By default, the j1Pub Component Catalog contains a Schematic Macro implementation for the j1FAT32 component. This macro contains an already named j1FAT32 library with the default configuration settings.

j1FAT32
j1FAT32 v1.0
www.j1sys.com

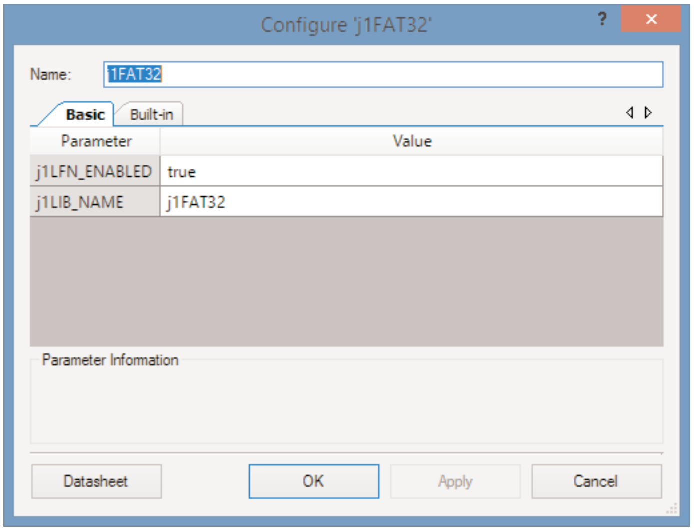
Component Configuration

Drag a j1FAT32 component onto the design. Double click the component symbol to open the Configure dialog.

The following sections describe the j1FAT32 parameters, and how they are configured using the Configure dialog.

Configure Tab

The Configure tab contains basic parameters for the j1FAT32 component. These parameters are the first ones that appear when you open the Configure dialog.



Configure 'j1FAT32'

Name: j1FAT32

Parameter	Value
j1LFN_ENABLED	true
j1LIB_NAME	j1FAT32

Configuration Parameters

Each configuration parameter is described below.

j1LFN_ENABLED

Description:	Boolean enables the inclusion of the optional long file name features.
Values:	true or false, default true

j1LIB_NAME

Description:	Libary name for all functions and definitions. All functions and definitions are prefixed by this value. Usually should be left as default.
Default Value:	j1FAT32

Component Defined Datatypes (typedef)

The following defined datatypes are part of the component definitions. They should be instantiated in the application code and passed to/from the API via pointers.

typedef	Description
j1FAT32_CTRL	Object control structure
j1FAT32_FCB	File Control Block for an opened file
j1FAT32_PART_TABLE_ENTRY	Partition Table Entry
j1FAT32_DIR_ENTRY	Directory Entry
j1FAT32_SEC_BFR	A union of varying sector formats used in FAT32

Application Programming Interface

Application Programming Interface (API) routines allow you to use the library at runtime. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name “j1FAT32_1” to the first instance of a component in a given design. We have included a macro that will name the instance “j1FAT32”. You can rename the instance to any unique value that follows the syntactic rules for identifiers. The instance name will not be used as a prefix of every global function name, variable, and constant symbol for the instance as a hardware component would commonly be handled. The $j1LIB_NAME parameter will be used as a prefix of every global function name, variable, and constant symbol for the instance. The default $j1LIB_NAME is “j1FAT32” and should normally not need to be changed. For readability, the $j1LIB_NAME used in the following table is “j1FAT32”.

Function	Description
j1FAT32_Start()	Initializes the control to be used with the functions.
j1FAT32_End()	Finalizes and clears the j1FAT32 object. Not currently needed for read-only.
j1FAT32_GetVolumeID()	Returns the Volume ID for the device.
j1FAT32_OpenDir()	Finds a directory based on a path and configures the object for subsequent directory entry reads.
j1FAT32_OpenDirCluster()	Opens a directory based on a starting cluster number and configures the object for subsequent directory entry reads.
j1FAT32_ReadNextDirEntry()	Reads next directory entry. Directory entries can be filtered based on optional matching filters.
j1FAT32_RewindDir()	Rewinds the currently opened directory to its beginning and configures the object for subsequent directory entry reads.
j1FAT32_CloseDir()	Finalizes and clears the directory reads. Not currently needed for read-only.
j1FAT32_OpenFile()	Opens a file based on a path and configures an FCB for subsequent file operations.
j1FAT32_OpenFileCluster()	Opens a file based on a starting cluster and configures an FCB for subsequent file operations.
j1FAT32_SeekFile()	Performs a seek on a previously opened FCB.
j1FAT32_ReadFile()	Performs a read on a previously opened FCB.
j1FAT32_RewindFile()	Performs a rewind on a previously opened FCB.
j1FAT32_CloseFile()	Finalizes and clears a previously opened FCB. Not currently needed for read-only.

Internal Functions and Utility Functions

Several other internal functions are exposed and could be used for advanced manipulation. There are also utility functions for manipulating file names.

Function	Description
j1FAT32_FollowFAT32()	Follow a FAT32 cluster chain.
j1FAT32_GetClusterStartLBA()	Get the start LBA for a cluster.
j1FAT32_ReadDeviceSector()	Read an LBA sector device relative.
j1FAT32_ReadVolumeSector()	Read an LBA sector volume relative
j1FAT32_NameToString()	Convert an 8.3 directory entry filename to a string.
j1FAT32_GetStartCluster()	Get the starting cluster from a directory entry.

API Details

bool j1FAT32_Start(j1FAT32_CTRL *f32C, bool (*readSector)(void *deviceCtrl, uint32 lba, uint8 *secBfr), void *deviceCtrl)

Description: Initializes the control to be used with the functions. This function must be called to initialize the object control before using any other function.

Parameters:

j1FAT32_CTRL	*f32C	Pointer to the object control structure
	*readSector	Pointer to the device readSector function
void	*deviceCtrl	Pointer to the device control structure

Return Value: true

bool j1FAT32_End(j1FAT32_CTRL *f32C)

Description: Finalizes and clears the j1FAT32 object. Not currently needed for read-only.

Parameters:

j1FAT32_CTRL	*f32C	Pointer to the object control structure

Return Value: true

bool j1FAT32_GetVolumeID(j1FAT32_CTRL *f32C, char *volumeIDBfr, uint8 bfrLen)

Description: Returns the Volume ID for the device.

Parameters:

j1FAT32_CTRL	*f32C	Pointer to the object control structure
char	*volumeIDBfr	Pointer to buffer to received Volume ID
uint8	bfrLen	Size of buffer for VolumeID

Return Value: true if successful, false if not

bool j1FAT32_OpenDir(j1FAT32_CTRL *f32C, char *dirPath)

Description: Finds a directory based on a path and configures the object for subsequent directory entry reads.

Parameters:

j1FAT32_CTRL	*f32C	Pointer to the object control structure
char	*dirPath	Pointer to directory path string

Return Value: true if successful, false if not

bool j1FAT32_OpenDirCluster(j1FAT32_CTRL *f32C, uint32 cluster)

Description: Opens a directory based on a starting cluster number and configures the object for subsequent directory entry reads.

Parameters:

j1FAT32_CTRL	*f32C	Pointer to the object control structure
uint32	cluster	32 bite cluster number

Return Value: None

bool j1FAT32_ReadNextDirEntry(j1FAT32_CTRL *f32C, j1FAT32_DIR_ENTRY *dirEntryBfr, char *lfnBfr, uint8 lfnBfrLen, char *match, uint8 matchAttrib, char *extMatch)

Description: Reads next directory entry. Directory entries can be filtered based on optional matching filters.

Parameters:

j1FAT32_CTRL	*f32C	Pointer to the object control structure
j1FAT32_DIR_ENTRY	*dirEntryBfr	Pointer to character array to ‘put’
char	*lfnBfr	Pointer to buffer for LFN
uint8	lfnBfrLen	Size of LFN buffer
char	*match	Match filename string
uint8	matchAttrib	Match Attrib bit mask
char	*extMatch	Match file extension string

Return Value: true if successful, false if not (end-of-directory)

bool j1FAT32_RewindDir(j1FAT32_CTRL *f32C)

Description: Rewinds the currently opened directory to its beginning and configures the object for subsequent directory entry reads.

Parameters:

j1FAT32_CTRL	*f32C	Pointer to the object control structure

Return Value: true

bool j1FAT32_CloseDir(j1FAT32_CTRL *f32C)

Description: Finalizes and clears the directory reads. Not currently needed for read-only.

Parameters:

j1FAT32_CTRL	*f32C	Pointer to the object control structure

Return Value: true

bool j1FAT32_OpenFile(j1FAT32_CTRL *f32C, char *fileSpec, j1FAT32_FCB *fcb)

Description: Opens a file based on a path and configures an FCB for subsequent file operations.

Parameters:	j1FAT32_CTRL	*f32C	Pointer to the object control structure
	char	*fileSpec	Pointer to fileSpec string
	j1FAT32_FCB	*fcb	Pointer to FCB to initialize and fill

Return Value: true if successful, false if not

bool j1FAT32_OpenFileCluster(j1FAT32_CTRL *f32C, uint32 cluster, uint32 fileSize, j1FAT32_FCB *fcb)

Description: Opens a file based on a starting cluster and configures an FCB for subsequent file operations.

Parameters:	j1FAT32_CTRL	*f32C	Pointer to the object control structure
	uint32	cluster	cluster start of file
	uint32	fileSize	file size in bytes
	j1FAT32_FCB	*fcb	Pointer to FCB to initialize and fill

Return Value: true if successful, false if not

bool j1FAT32_SeekFile(j1FAT32_FCB *fcb, uint32 recAddr)

Description: Performs a seek on a previously opened FCB.

Parameters:	j1FAT32_FCB	*fcb	Pointer to FCB for opened file
	uint32	recAddr	32 bit files seek offset

Return Value: true if successful, false if not

bool j1FAT32_ReadFile(j1FAT32_FCB *fcb, uint8 *bfrPtr, uint16 bfrSiz, uint16 *bfrRead)

Description: Performs a read on a previously opened FCB.

Parameters:	j1FAT32_FCB	*fcb	Pointer to FCB for opened file
	uint8	*bfrPtr	Pointer to read buffer to be filled
	uint16	bfrSiz	Buffer size (max bytes to read)
	uint16	*bfrRead	Pointer to variable to fill with read byte count

Return Value: true if successful, false if not

bool j1FAT32_RewindFile(j1FAT32_FCB *fcb)

Description: Performs a rewind on a previously opened FCB.

Parameters:	j1FAT32_FCB	*fcb	Pointer to the FCB of opened file

Return Value: true if successful, false if not

bool j1FAT32_CloseFile(j1FAT32_FCB *fcb)

Description: Finalizes and clears a previously opened FCB. Not currently needed for read-only.

Parameters:	j1FAT32_FCB	*fcb	Pointer to the FCB of opened file

Return Value: true if successful, false if not

uint32 j1FAT32_FollowFAT32(j1FAT32_CTRL *f32C, uint32 cluster)

Description: Follow a FAT32 cluster chain.

Parameters:	j1FAT32_CTRL	*f32C	Pointer to the object control structure
	uint32	cluster	current cluster number

Return Value: next cluster number, zero if end of cluster chain

uint32 j1FAT32_GetClusterStartLBA(j1FAT32_CTRL *f32C, uint32 cluster)

Description: Get the start LBA for a cluster.

Parameters:	j1FAT32_CTRL	*f32C	Pointer to the object control structure
	uint32	cluster	current cluster number

Return Value: LBA of first sector in cluster

bool j1FAT32_ReadDeviceSector(j1FAT32_CTRL *f32C, uint32 lba, uint8 *secBfr)

Description: Read an LBA sector device relative.

Parameters:	j1FAT32_CTRL	*f32C	Pointer to the object control structure
	uint32	lba	device LBA
	uint8	*secBfr	Pointer to 512 byte sector buffer

Return Value: true if successful, false if not

bool j1FAT32_ReadVolumeSector(j1FAT32_CTRL *f32C, uint32 lba, uint8 *secBfr)

Description: Read an LBA sector volume relative.

Parameters:	j1FAT32_CTRL	*f32C	Pointer to the object control structure
	uint32	lba	volume LBA
	uint8	*secBfr	Pointer to 512 byte sector buffer

Return Value: true if successful, false if not

bool j1FAT32_NameToString(j1FAT32_DIR_ENTRY *dirEntry, char *nameStringBfr, uint8 nameStringBfrLen)

Description: Convert an 8.3 directory entry filename to a string.

Parameters:

j1FAT32_DIR_ENTRY	*dirEntry	Directory entry buffer to convert
char	*nameStringBfr	Pointer to string buffer to convert into
uint8	nameStringBfrLen	Size of nameStringBfr

Return Value: true if successful, false if not

uint32 j1FAT32_GetStartCluster(j1FAT32_DIR_ENTRY *dirEntry)

Description: Get the starting cluster from a directory entry.

Parameters:

j1FAT32_DIR_ENTRY	*dirEntry	Directory entry buffer

Return Value: first cluster number in file