

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



ECSD API Reference

Version 2.0.0

© Copyright 2014 EmbCode AB

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



Table of contents

1	GENERAL OPERATIONS (ECSD_*)	3
1.1	ECSD_INIT	3
1.2	ECSD_CONNECTTOCARD.....	4
2	BLOCK DEVICE OPERATIONS (ECSD_*)	5
2.1	ECSD_READSECTOR	5
2.2	ECSD_WRITESECTOR.....	6
2.3	ECSD_GETVOLUMEINFORMATION	7
3	LOW LEVEL SPI DRIVER OPERATIONS (ECSD_LOWLEVEL_*)	8
3.1	ECSD_LOWLEVEL_INIT	8
3.2	ECSD_LOWLEVEL_ISCARDPRESENT	9
3.3	ECSD_LOWLEVEL_ISCARDWRITEPROTECTED	10
3.4	ECSD_LOWLEVEL_SHIFTDATA.....	11
3.5	ECSD_LOWLEVEL_READBLOCK	13
3.6	ECSD_LOWLEVEL_WRITEBLOCK.....	14
3.7	ECSD_LOWLEVEL_RELEASECS	15

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



1 General operations (ECSD_*)

1.1 ECSD_Init

The ECSD_Init function initialises ECSD and calls the low-level driver to initialize it.

```
ECSD_ErrorCode ECSD_Init(void);
```

Parameters

None

Return value

Returns one of the ECSD error codes (ECSD_STATUS_OK on success).

Remarks

Note: This function does not start any communication with the SD card. It is just called to initialise e.g. I/O pins and other hardware necessary to later communicate with the SD card.

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



1.2 ECSD_ConnectToCard

The ECSD_ConnectToCard function attempts to connect to the SD card.

```
ECSD_ErrorCode ECSD_ConnectToCard(void) ;
```

Parameters

None

Return value

Returns one of the ECSD error codes (ECSD_STATUS_OK on success).

Remarks

If ECSD_ConnectToCard returns ECSD_STATUS_OK you can use ECSD_ReadSector, ECSD_WriteSector and ECSD_GetVolumeInformation to access the data on the card.

See also

ECSD_ReadSector, ECSD_WriteSector, ECSD_GetVolumeInformation

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	

2 Block device operations (ECSD_*)

2.1 ECSD_ReadSector

The ECSD_ReadSector function reads a sector (block) from the connected SD card.

```
ECSD_ErrorCode ECSD_ReadSector(  
    struct ECF_BlockDriver *pBlockDriver,  
    DWORD sector,  
    BYTE *data,  
    BYTE flags  
);
```

Parameters

pBlockDriver

A pointer to an ECF_BlockDriver struct for EcFAT compatibility. If you are not using EcFAT you can safely pass NULL.

sector

The sector to read.

data

A pointer to a buffer that can hold a block of the sector size. Call ECSD_GetVolumeInformation to obtain the sector size.

flags

Flags that ECF will pass when reading sectors. It is safe to pass 0.

Return value

Returns one of the ECSD error codes (ECSD_STATUS_OK on success).

See also

ECSD_GetVolumeInformation

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



2.2 ECSD_WriteSector

The ECSD_LowLevel_Init function initialises the hardware necessary to communicate with the SD card.

```
ECSD_ErrorCode ECSD_WriteSector(  
    struct ECF_BlockDriver *pBlockDriver,  
    DWORD sector,  
    BYTE *data,  
    BYTE flags  
);
```

Parameters

pBlockDriver

A pointer to an ECF_BlockDriver struct for EcFAT compatibility. If you are not using EcFAT you can safely pass NULL.

sector

The sector to write.

data

A pointer with a buffer of the data to write. The buffer must hold an entire sector. Call ECSD_GetVolumeInformation to obtain the sector size.

flags

Flags that ECF will pass when reading sectors. It is safe to pass 0.

Return value

Returns one of the ECSD error codes (ECSD_STATUS_OK on success).

See also

ECSD_GetVolumeInformation

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



2.3 ECSD_GetVolumeInformation

The ECSD_GetVolumeInformation function gives information about the disk and sector size of the connected SD card.

```
ECSD_ErrorCode ECSD_GetVolumeInformation(  
    struct ECF_BlockDriver *pBlockDriver,  
    WORD *pwSectorSize,  
    DWORD *pdwNumberOfSectors  
);
```

Parameters

pBlockDriver

A pointer to an ECF_BlockDriver struct for EcFAT compatibility. If you are not using EcFAT you can safely pass NULL.

pwSectorSize

A pointer to WORD that will receive the sector size. You may pass NULL if you don't need the sector size.

pdwNumberOfSectors

A pointer to a DWORD that will receive the number of sectors. You may pass NULL if you don't need the sector size.

Return value

Returns one of the ECSD error codes (ECSD_STATUS_OK on success).

Remarks

You need to call ECSD_ConnectToCard before calling ECSD_GetVolumeInformation.

See also

ECSD_ConnectToCard

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



3 Low level SPI driver operations (ECSD_LowLevel_*)

3.1 ECSD_LowLevel_Init

The ECSD_LowLevel_Init function initialises the hardware necessary to communicate with the SD card.

```
ECSD_ErrorCode ECSD_LowLevel_Init(void);
```

Parameters

None

Return value

Returns one of the ECSD error codes (ECSD_STATUS_OK on success).

Remarks

Note: You should never call this function directly. The API documentation is present to allow you to write your own low level driver optimized for your hardware.

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



3.2 ECSD_LowLevel_IsCardPresent

The ECSD_LowLevel_IsCardPresent function checks if a card is inserted into the SD Card holder.

```
BOOL ECSD_LowLevel_IsCardPresent(void);
```

Parameters

None

Return value

TRUE if a card is detected in the slot, FALSE if not.

Remarks

If you are communicating with an SD Card that is fixed to the system, always return TRUE.

Note: You should never call this function directly. The API documentation is present to allow you to write your own low level driver optimized for your hardware.

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



3.3 ECSD_LowLevel_IsCardWriteProtected

The ECSD_LowLevel_IsCardWriteProtected function checks if an inserted card is write protected by its write protected tab.

```
BOOL ECSD_LowLevel_IsCardWriteProtected(void);
```

Parameters

None

Return value

TRUE if the card is write protected, FALSE if not.

Remarks

If you are communicating with an SD Card that is fixed to the system or through a holder that doesn't support the write protected tab, always return FALSE.

Note: You should never call this function directly. The API documentation is present to allow you to write your own low level driver optimized for your hardware.

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



3.4 ECSD_LowLevel_EnableFastClock

The ECSD_LowLevel_EnableFastClock enables or disables the fast SPI clock.

```
void ECSD_LowLevel_EnableFastClock (  
    BOOL bEnable  
);
```

Parameters

bEnable

During startup and card detection a slow clock is used for compatibility with MMC cards.

If called with FALSE, the slow clock should be enabled for SPI. The slow clock can not be faster than 400 kHz.

If called with TRUE, the fast clock should be enabled for SPI. The fast clock can not be faster than 25 MHz (20MHz if you plan to use MMC cards).

Note: You should never call this function directly. The API documentation is present to allow you to write your own low level driver optimized for your hardware.

Return value

No return value.

Remarks

This must be called before accessing files on the disk.

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



3.5 ECSD_LowLevel_ShiftData

The ECSD_LowLevel_ShiftData function shifts a byte through to and from the card.

```
BYTE ECSD_LowLevel_ShiftData(  
    BYTE bData  
);
```

Parameters

bData

This is the data to shift out.

Return value

The data shifted in from the card.

Remarks

The low level driver must activate (lower) CS (Chip-Select) when this function is called if it is not already activated.

Note: You should never call this function directly. The API documentation is present to allow you to write your own low level driver optimized for your hardware.

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



3.6 ECSD_LowLevel_ReadBlock

The ECSD_LowLevel_ReadBlock function reads a block of data from the SPI bus.

```
ECSD_ErrorCode ECSD_LowLevel_ReadBlock (  
    BYTE *data,  
    WORD size  
);
```

Parameters

data

This is the pointer to the buffer where the data should be placed.

size

The number of bytes to read.

Return value

Returns one of the ECSD error codes (ECSD_STATUS_OK on success).

Remarks

Called to read a block of data from the SPI bus. 0xFF should be shifted out on the bus while the data is being read.

Note: You should never call this function directly. The API documentation is present to allow you to write your own low level driver optimized for your hardware.

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



3.7 ECSD_LowLevel_WriteBlock

The ECSD_LowLevel_WriteBlock writes a block of data to the SPI bus.

```
ECSD_ErrorCode ECSD_LowLevel_WriteBlock(  
    BYTE *data,  
    WORD size  
);
```

Parameters

data

The pointer to the buffer with the data to write.

size

The number of bytes to write.

Return value

Returns one of the ECSD error codes (ECSD_STATUS_OK on success).

Remarks

Called to write data to the SPI bus. Read data should be discarded.

Note: You should never call this function directly. The API documentation is present to allow you to write your own low level driver optimized for your hardware.

Document name: ECSD API Reference	Version 2.0.0
Internal reference: Products/ECSD/API Reference/3548	



3.8 ECSD_LowLevel_ReleaseCS

The ECSD_LowLevel_ReleaseCS de-activates the CS line (pull it up).

```
void ECSD_LowLevel_ReleaseCS(void);
```

Parameters

None

Return value

No return value.

Remarks

Note: You should never call this function directly. The API documentation is present to allow you to write your own low level driver optimized for your hardware.