

# Nagyteljesítményű mikrovezérlők

## SD kártyák és FAT file rendszerek

Scherer Balázs



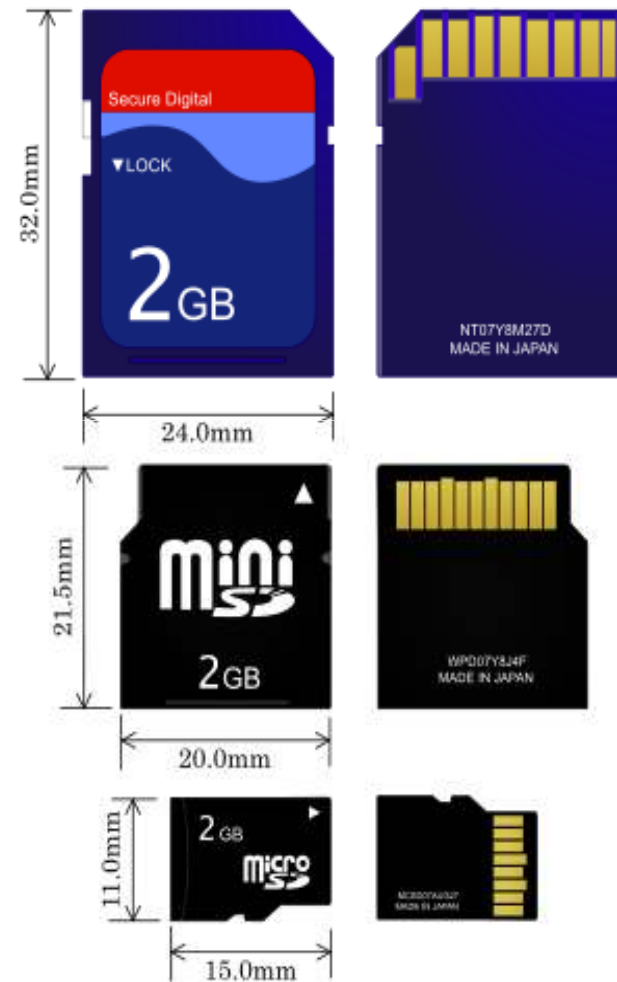
Méréstechnika és  
Információs Rendszerek  
Tanszék

# SD kártyák megjelenése

- 1999-ben egyezett meg a SanDisk, Matsushita és Toshiba, hogy létrehozza a 24 mm × 32 mm × 2.1 mm méretű SD (Secure Digital) Memory Card-ot.
  - A kártyák 2000 óta elérhetőek a piacon
- A Standard SD kártyáknak max 2 GB-os kapacitása van.
  - Ezt bővítik ki az SDHC (high-capacity) kártyák 4GB feletti méretekre
  - A 2009-es új SDXC (eXtended Capacity) szabvány már 2 TerraB kapacitást is megenged.
- Az egyes szabvány interfészek között vannak különbségek (a fizikai méretek azonosak), amik gondot okozhatnak.

# SD kártya típusok

- SD card 32mm x 24mm
- MiniSD card 21,5mm x 20mm
- MicroSD card 15mm x 11mm



# SD kártya sebességek

- Az SD kártyák sebességei:
  - Az SD Association által specifikált minimális sebesség 8Mbit/sec
- A jelenleg forgalomba tévő kártyák
  - Class 2: 16 MBit/s (2 MByte/s), 13x
  - Class 4: 32 MBit/s (4 MByte/s), 26x
  - Class 6: 48 MBit/s (6 MByte/s), 40x

A sebességet sokszor "X" rating-ben adják meg, ami a standard CD-ROM 1.2Mbit/s-es sebességéhez viszonyít.

# SD kártyák kezelése

- Kommunikációs módok
  - **One-bit SD mode:** Különálló parancs és adat csatorna.
  - **Four-bit SD mode:** Extra adatlábak.
  - **SPI mode:** Egyszerűsített kommunikációs elsősorban mikrovezérlők részére.
- Az összes kártyának támogatnia kell ezeket a módokat kivéve a microSD-t ahol az SPI mód opcionális.
- Az SD kártyák normál esetben hasonlóan a HDD-khez Sector-Block felosztásuak.
  - A block: hány byte írható, olvasható egyszerre a blokkos adatátvitelnél.
    - A tipikus block méret 512byte szokott lenni.
  - A sector pedig azt jelenti, hogy hány blokk törölhető egyszerre.

Ezek a paraméterek kiolvashatóak az adott kártya információs regiszteréből. A legtöbb kártya a normál data area-n kívül még rendelkezik protected area-val is a bizalmas adatok számára.

# SD kártya alapregiszterek

- **OCR (Operation Condition Register):** a kártya működési feszültség tartományát adja meg (tipuksan 2.7V-3.6V)
- **CID (Card Identification Register):** 16byte-os egyedi azonosító, ami a Manufacturer, OEM/Application, Product Name, Product Revision, Serial Number, Manufacture Date Code, CRC7 checksum.
- **Card Specific Data (CSD):** Ez az adatregiszter tartalmaz minden a kártya kezelésével kapcsolatos felhasználói információkat:
  - data read access-time (pl.: 1ms)
  - max. data transfer rate (pl.: 25MHz)
  - max. read data block length (512byte)
  - max. write data block length (512byte)
  - partial blocks for read allowed (Yes)
  - device size
  - max. read current
  - erase single block enable
  - erase sector size (pl.: 32 blocks)
  - write speed factor (pl.: X16)
- **SRC (SD CARD Configuration register):** Ez a regiszter tartalmazza az adott SD kártyára jellemző speciális információkat (általában security supporthoz tartozó dolgok).
- **RCA register:** Címregiszter a kártya azonosítására

# SPI mód

- **Legegyszerűbb átvitel**
  - Korlátozott sebesség 1bit adat
  - STM32 SPI SCK frekvencia max. 18MHz

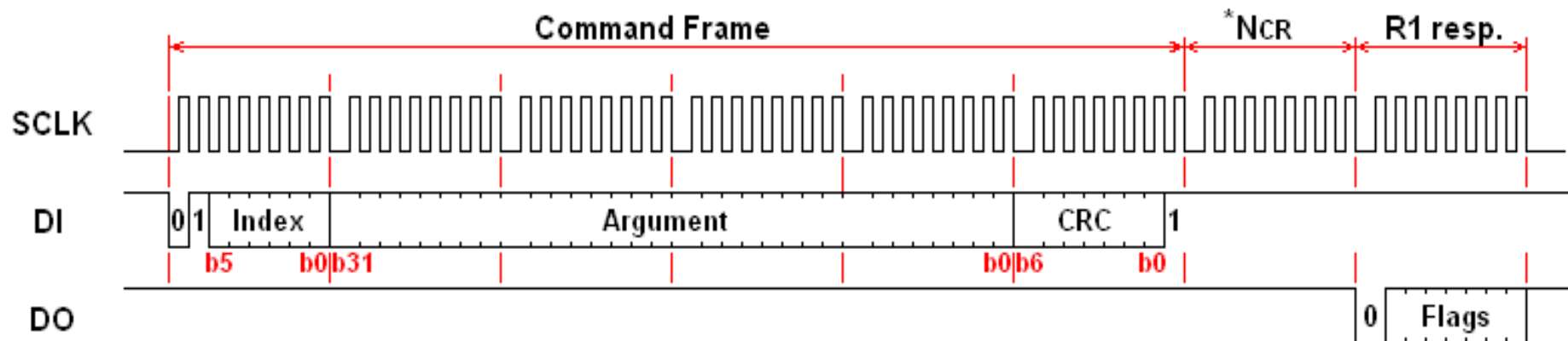
Table 3-2. SPI Bus Mode Pad Definition

Pin #	Name	Type <sup>1</sup>	SPI Description
1	CS	I	Chip Select (Active low)
2	DataIn	I	Host to Card Commands and Data
3	VSS1	S	Supply Voltage Ground
4	VDD	S	Supply Voltage
5	CLK	I	Clock
6	VSS2	S	Supply Voltage Ground
7	DataOut	O	Card to Host Data and Status
8	RSV(2)	I	Reserved
9	RSV(2)	I	Reserved

# SPI kommunikációs frame

Az SPI kommunikációs frame 6 byte-ból áll. A CRC bekapcsolható és kikapcsolható.

Byte 1				Bytes 2—5				Byte 6		
7	6	5	0	31			0	7	0	
0	1	Command		Command Argument				CRC		1





# SPI kommunikációs parancsok

A parancsok a parancs kód +0x40 értékkel küldődnek el az SPI buszon.

Alap parancsok:

CMD INDEX	SPI Mode	Argument	Resp	Abbreviation	Command Description
CMD0	Yes	None	R1	GO_IDLE_STATE	Resets the SD Card
CMD1	Yes	None	R1	SEND_OP_COND	Activates the card's initialization process.

Identifikációs parancsok:

CMD9	Yes	None	R1	SEND_CSD	Asks the selected card to send its card-specific data (CSD).
CMD10	Yes	None	R1	SEND_CID	Asks the selected card to send its card identification (CID).

Adatátvitel leállítás és státusz parancsok:

CMD12	Yes	None	R1b	STOP_TRANSMISSION	Forces the card to stop transmission during a multiple block read operation.
CMD13	Yes	None	R2	SEND_STATUS	Asks the selected card to send its status register.

# SPI kommunikációs parancsok

A parancsok a parancs kód +0x40 értékkel küldődnek el az SPI buszon.

## Olvasó parancsok

CMD16	Yes	[31:0] block length	R1	SET_BLOCKLEN	Selects a block length (in bytes) for all following block commands (read & write). <sup>1</sup>
CMD17	Yes	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command. <sup>2</sup>
CMD18	Yes	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a STOP_TRANSMISSION command.

## Író parancsok

CMD24	Yes	[31:0] data address	R1 <sup>3</sup>	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command. <sup>4</sup>
CMD25	Yes	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a stop transmission token is sent (instead of 'start block').

# SPI kommunikációs parancsok

A parancsok a parancs kód +0x40 értékkel küldődnek el az SPI buszon.

## Törlést kijelölő parancsok

CMD32	Yes	[31:0] data address	R1	ERASE_WR_BLK_START_ADDR	Sets the address of the first write block to be erased.
CMD33	Yes	[31:0] data address	R1	ERASE_WR_BLK_END_ADDR	Sets the address of the last write block in a continuous range to be erased.

## Törlő parancsok

CMD38	Yes	[31:0] don't care*	R1b	ERASE	Erases all previously selected write blocks.
-------	-----	--------------------	-----	-------	--

# SPI kommunikációs parancsok

A parancsok a parancs kód +0x40 értékkel küldődnek el az SPI buszon.

## Speciális parancsok

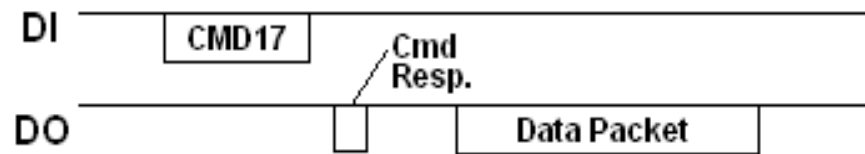
CMD27	Yes	None	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.
CMD28 <sup>1</sup>	Yes	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29 <sup>4</sup>	Yes	[31:0] data address	R1b	CLR_WRITE_PROT	If the card has write protection features, this command clears the write protection bit of the addressed group.
CMD30	Yes	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card has write protection features, this command asks the card to send the status of the write protection bits. <sup>2</sup>

# SPI mód inicializáció

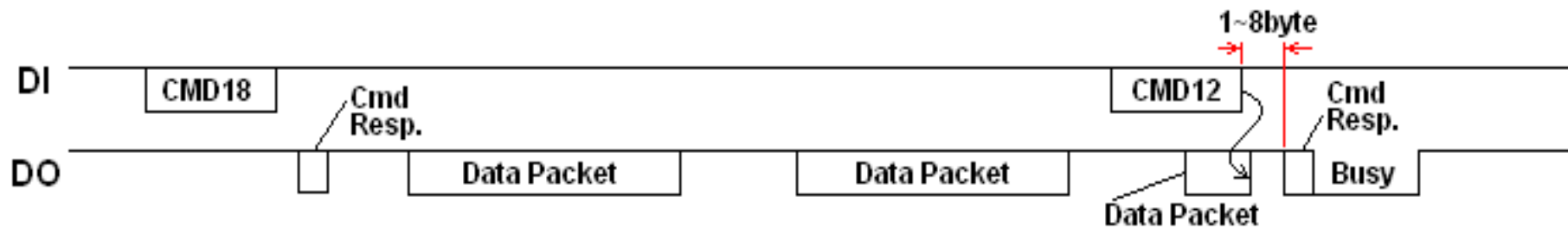
- Az SD kártyák alapvetően SD módban indulnak el
  - először is SPI módba kell azokat rakni
  - Reszet parancs alatt a CS lábat le kell húzni.
  - Bár az SPI módban a CRC védelem ki van kapcsolva, az első reszet parancsnál még SD módban van a kártya tehát érvényes CRC mezőt kell átküldeni
    - (mivel a parancsnak nincs aktívan változó paramétere, ezért az egész reszet parancs kezelhető egy byte-sorozatnak 0x40, 0x0, 0x0, 0x0, 0x0, 0x95).
  - Ennél a parancsnál az SPI frekvencia nem haladhatja meg a 400kHz-et.

# Olvasás SPI módban

Az SPI mód egy blokk (CMD17) és több blokk (CMD18) egyszerre való olvasását is támogatja. A Data packetben lévő CRC-t mindenképpen ki kell olvasni, akkor is ha nem használjuk.

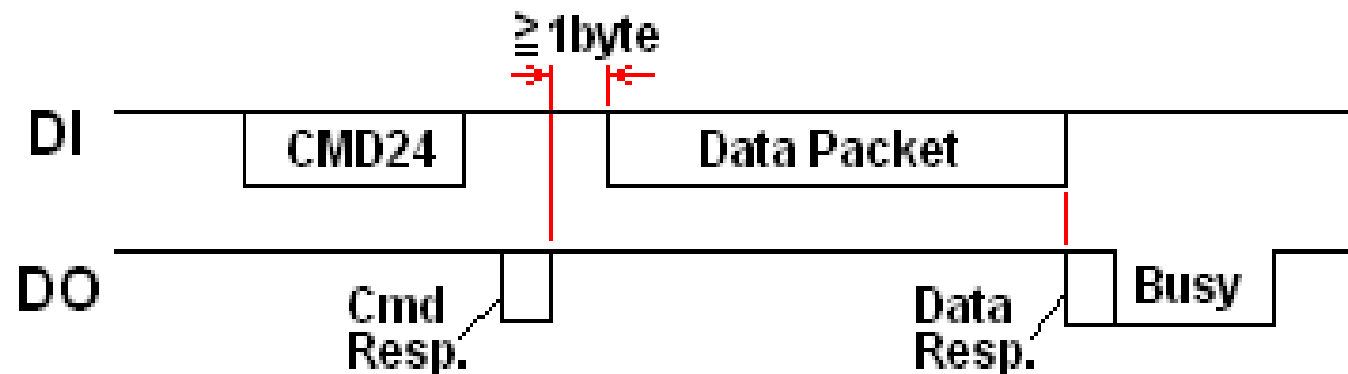


Több blokk esetében a Stop transmission paranccsal (CMD12) le lehet állítani az adatátvitelt.



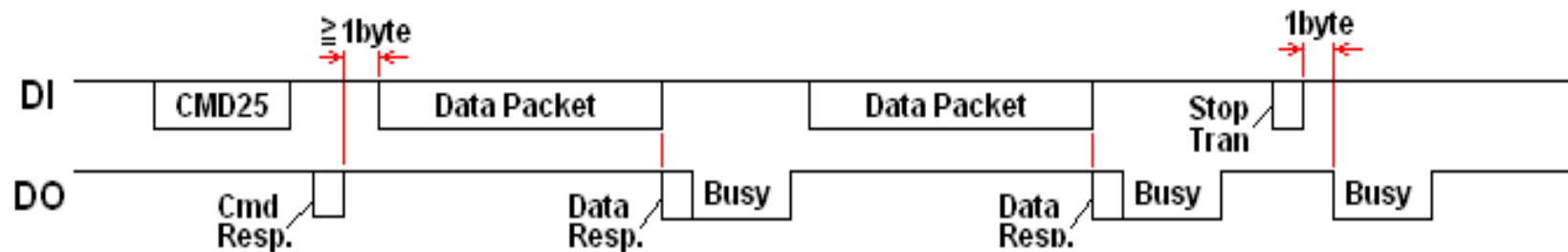
# Írás SPI módban

Az SPI blokk támogatja az egy és a több blokkos írást is (CMD24, CMD25). Mindegyik blokk írás egy egy byte-os Start Block tokennel indul. Az adat megérkezése után az SD kártya küld egy data\_response tokenet, majd amíg az adatokat ténylegesen kiírja a Flash memóriába folyamatosan busy tokenet küld a buszra (lent tartja DataOut lábát). Az írás véget érését vagy ennek a busy jelenek a figyelésével, vagy a státusz információ kiolvasásával (CMD13) tudja a hosszt megállapítani.



# Írás SPI módban, több byte

Több byte-os írása esetében a befejezésnél a Start block token helyett egy egy byte-os Stop Tran tokenet kell küldeni.





# FAT file rendszer

- A FAT első verziója a FAT12-t (32 Mbyte) 1980-ban fejlesztették ki floppy lemezek számára.
- A következő verzió a FAT16-volt (2 Gbyte), ami 1987-ben látott napvilágot.
- Az utolsó a FAT32-volt, ami 1996-ban jelent meg (2 Tbyte), a Windows-on a SCANDISK alkalmazás 16 bites szektor számlálója miatt volt egy kb. 130 Gbyte-os határ.

# FAT file rendszer felépítése

<b>Contents</b>	Boot Sector	FS Information Sector (FAT32 only)	More reserved sectors (optional)	File Allocation Table #1	File Allocation Table #2	Root Directory (FAT12/16 only)	Data Region (for files and directories) ... (To end of partition or disk)
<b>Size in sectors</b>	(number of reserved sectors)			(number of FATs)* (sectors per FAT)		(number of root entries*32)/Bytes per sector	NumberOfClusters*SectorsPerCluster

## ■ **Boot Sector: (Particion Boot Record):**

- általában az operációs rendszer bootload-erjét tartalmazza.
- A lefoglalt terület mérete a Boot sector egy mezője által azonosítódik.
- Nem mindig a boot sector az első szektor a disk-en.
- Particionált egységeknél az első szektor a master boot record, nem particionált egységeknél a Volume boot record.
- Az első 36 byte struktúrája minden FAT file rendszer esetében azonos
  - Ez tartalmaz egy jump vetort ha innen indulunk az itt található címre ugrik a programvégrehajtás. Tartalmazza még az OEM nevét (mire formázták) a szektoronkénti byte-ok számát (ált 512), a clusterenkénti sector-ok számát (2 1-128-ig terjedő hatványa, max 32k byte/ cluster), valamint az összes sector számát.

# FAT file rendszer felépítése

<b>Contents</b>	Boot Sector	FS Information Sector (FAT32 only)	More reserved sectors (optional)	File Allocation Table #1	File Allocation Table #2	Root Directory (FAT12/16 only)	Data Region (for files and directories) ... (To end of partition or disk)
<b>Size in sectors</b>	(number of reserved sectors)			(number of FATs)* (sectors per FAT)		(number of root entries*32)/Bytes per sector	NumberOfClusters*SectorsPerCluster

- **FS Information sector (csak FAT32):** A FAT32-ben mutatták be elsősorban a szabad terület gyors nyilvántartására.
- **FAT (File Allocation Table):** A partíciók egyenlő méretű clusterekre vannak bontva, a cluster méret függhet az alkalmazott FAT file rendszertől és a partíció méretétől. Általában a 2k és a 32k közötti méreteket preferálják. Minden file méretétől függően egy, vagy több ilyen clustert foglal el. Egy file a clusterek láncolt listájával megadható, bár sokszor még az egy file hoz tartozó clusterek se mindig egymás mellett találhatóak: fragmentálódik a file.

# FAT tábla

A FAT egy leíró lista, amely egy térképet ad a partícióban található clusterek-hez a FAT16 és a FAT32 esetében a leíró tábla minden egyes sector-hoz 32bitet tartalmaz (a FAT tábla mérete függ a sectorok számától). A leíró lista az alábbi bejegyzéseket tartalmazhatja:

- A következő cluster száma
- Speciális *end of clusterchain (EOC)* jelzés a lánc végén, a file utolsó clusterere.
- Speciális a bad cluster jelzés
- Speciális jelzés a reserved cluster számára
- A 0 ha a cluster nem használt.

FAT12	FAT16	FAT32	Description
0x000	0x0000	0x00000000	Free Cluster
0x001	0x0001	0x00000001	Reserved value; do not use
0x002–0xFE7	0x0002–0xFFEF	0x00000002–0x0FFFFFFE	Used cluster; value points to next cluster
0xFF0–0xFF6	0xFFF0–0xFFF6	0xFFFFFFFF0–0xFFFFFFFF6	Reserved values; do not use <sup>[30]</sup>
0xFF7	0xFFF7	0xFFFFFFFF7	Bad sector in cluster or reserved cluster
0xFF8–0xFFF	0xFFF8–0xFFFF	0xFFFFFFFF8–0xFFFFFFFFF	Last cluster in file

# Directory tábla

A directory table egy speciális file. Minden directory, vagy file, ami benne van egy 32byte-os blokkal azonosítódik. Mindegyik blokk tartalmazza a következőket:

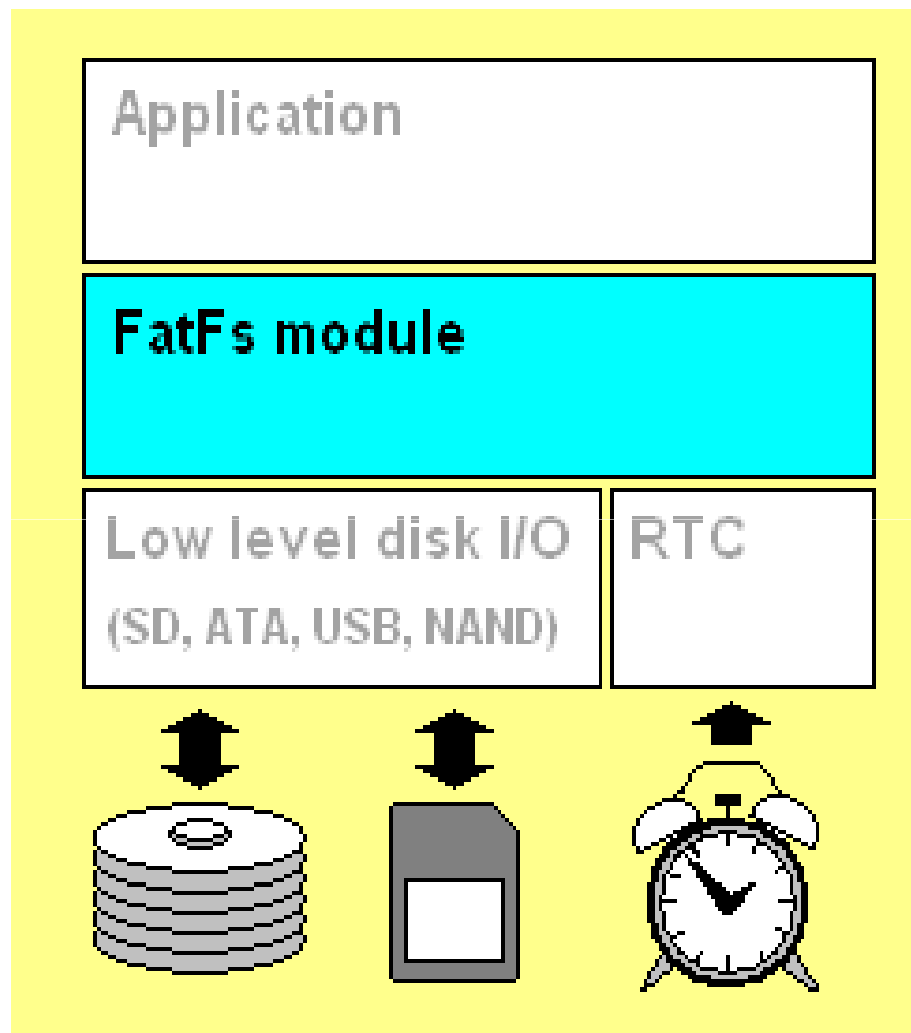
- Nevet
- Kiterjesztést
- A file tulajdonságát: read only hidden stb
- A létrehozás idejét.
- Az utolsó hozzáférés idejét
- A file, vagy directory első clusterének címét
- A file méretét byte-okban.

A FAT16 és a FAT12 esetében van egy kitüntetett root directory rész, míg a FAT32 estében minden directory a DATA részben tárolódik. A hosszú file nevek egy trükkel plusz bejegyzésként adódnak hozzá minden egyes file bejegyzés elején.

# Chan FatFS

- Kifejezetten beágyazott rendszerekhez létrehozott FAT file-rendszer:
- ANSI C-ben íródott
- Platform független
- FAT sub-types: FAT12, FAT16 and FAT32.
- Nyitott file-ok száma: Nem limitált a rendelkezésre álló memóriától függ
- File méret: A FAT specifikációtól függően 4G-1 bytes.
- Cluster méret: 64kbytes, vagy 32kbytes.
- Sector méret: FAT specifikáció függő (max. 4K bytes).

# Chan FatFS felépítés



# Chan FatFS portolás

Function	Required when:
<code>disk_initialize</code>	Always
<code>disk_status</code>	Always
<code>disk_read</code>	Always
<code>disk_write</code>	<code>_FS_READONLY == 0</code>
<code>disk_ioctl (CTRL_SYNC)</code>	<code>_FS_READONLY == 0</code>
<code>disk_ioctl (GET_SECTOR_COUNT)</code>	<code>USE_MKFS == 1</code>
<code>disk_ioctl (GET_SECTOR_SIZE)</code>	<code>MAX_SS &gt;= 1024</code>
<code>disk_ioctl (GET_BLOCK_SIZE)</code>	<code>USE_MKFS == 1</code>
<code>get_fattime</code>	<code>FS_READONLY == 0</code>

- `disk_initialize` – Disk drive inicializáció
- `disk_status` – Disk drive státusz lekérdezés
- `disk_read` – Sector olvasás
- `disk_write` - Sector írás
- `disk_ioctl` – Disk specifikus tulajdonságok
- `get_fattime` – Rendszer idő (naptári óra) lekérdezés



# Chan FatFS felhasználói interfész

- [f\\_mount](#) - Register/Unregister a Work Area
- [f\\_open](#) - Open/Create a File
- [f\\_close](#) - Close a File
- [f\\_read](#) - Read File
- [f\\_write](#) - Write File
- [f\\_lseek](#) - Move R/W Pointer, Expand File Size
- [f\\_truncate](#) - Truncate File Size
- [f\\_sync](#) - Flush Cached Data
- [f\\_opendir](#) - Open a Directory
- [f\\_readdir](#) - Read a Directory Item
- [f\\_getfree](#) - Get Free Clusters
- [f\\_stat](#) - Get File Status

# Chan FatFS felhasználói interfész

- [f\\_mkdir](#) - Create a Directory
- [f\\_unlink](#) - Remove a File or Directory
- [f\\_chmod](#) - Change Attribute
- [f\\_utime](#) - Change Timestamp
- [f\\_rename](#) - Rename/Move a File or Directory
- [f\\_mkfs](#) - Create a File System on the Drive
- [f\\_forward](#) - Forward file data to the stream directly
- [f\\_chdir](#) - Change current directory
- [f\\_chdrive](#) - Change current drive
- [f\\_gets](#) - Read a string
- [f\\_putc](#) - Write a character
- [f\\_puts](#) - Write a string
- [f\\_printf](#) - Write a formatted string

# Chan FatFS méret

## Memory Usage (R0.07e)

	AVR	H8/300H	PIC	TLCS-870/C	V850ES	SH2	ARM7TDMI	x86
Compiler	WinAVR (gcc)	CH38	C30 (gcc)	CC870C	CA850	SHC	WinARM (gcc)	VC6
<code>_WORD_ACCESS</code>	1	0	0	1	1	0	0	1
text (Full, R/W)	12194	10559	10924	15229	7686	8727	10564	7342
text (Min, R/W)	7988	6903	7108	9960	4884	5651	6544	4764
text (Full, R/O)	5532	4753	5020	6760	3462	3777	4624	3316
text (Min, R/O)	4040	3631	3736	5083	2556	2907	3284	2510
bss	$D*2 + 2$	$D*4 + 2$	$D*2 + 2$	$D*2 + 2$	$D*4 + 2$	$D*4 + 2$	$D*4 + 2$	$D*4 + 2$
Work area ( <code>_FS_TINY == 0</code> )	$D*560 +$ $F*544$	$D*560 +$ $F*550$	$D*560 +$ $F*544$		$D*560 +$ $F*550$	$D*560 +$ $F*550$	$D*560 +$ $F*550$	$D*560 +$ $F*550$
Work area ( <code>_FS_TINY == 1</code> )	$D*560 +$ $F*32$	$D*560 +$ $F*36$	$D*560 +$ $F*32$	$D*560 +$ $F*32$	$D*560 +$ $F*36$	$D*560 +$ $F*36$	$D*560 +$ $F*36$	$D*560 +$ $F*36$

- D: a kötetek száma
- F: nyitott file-ok száma