

# ARM Cortex magú mikrovezérlők

## Universal Serial Bus

Scherer Balázs



Méréstechnika és  
Információs Rendszerek  
Tanszék

# Célok

- Olcsó, egységes interface PC perifériákhoz – eredetileg csak kis sebességre (egér, billentyűzet, joystick...)
- Kevés vezeték, a tápfeszültség is jusson át az eszközhöz
- Tetszőleges számú eszköz csatlakoztatása
- Egyszerű használat → Plug & Play, dinamikusan töltődő driver-ek

## Fejlesztés kezdete:

1994 Intel, Microsoft, IBM, Compaq, NEC

Később sokan csatlakoztak: USB IF – USB Implementers Forum

# Szabványok, verziók

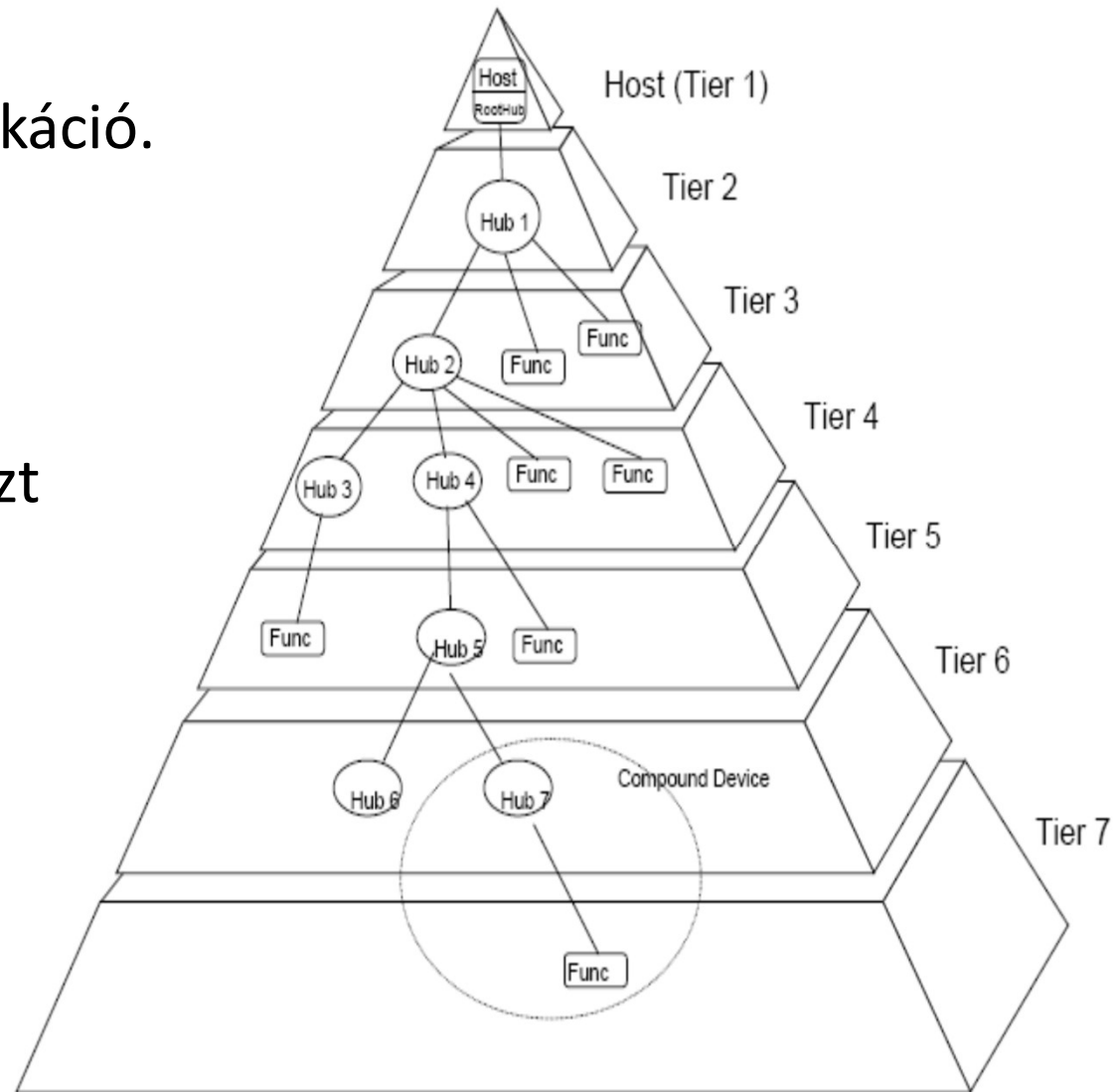
- 1996 USB 1.0 (nem terjedt el)
  - Még nem támogatja a Hub-okat
- 1998 USB 1.1
  - Low speed: 1.5 Mbit/sec
  - Full speed: 12 Mbit/sec
- 2001 USB 2.0
  - High speed: 480 Mbit/sec
  - Low-/High-power: 100 mA/500 mA

# Szabványok, verziók

- 2008 USB 3.0
  - Super-Speed: 5Gbit/sec (4 Gbit/sec effektív)
  - Low/high power mode: 150 mA/900 mA
  - Battery charger mode (kommunikáció nélkül) 1500 mA
  - Eszközök megjelenése: 2010
  - OS támogatás: Linux, Windows 8
- 2013 USB 3.1
  - Sebesség növelése 10 Gbit/sec-re
- 2017 USB 3.2
  - Támogatja a USB 3.2 Gen 1x2 és USB 3.2 Gen 2x2 multi-link módokat.

# Architektúra

- „Többszintű csillag” (tired star) 3-7 réteg (root + max 6 réteg)
- Master – Slave alapú kommunikáció. Hub csak továbbít.
- Hoszt felől jövő üzeneteket mindenki látja
- A perifériák válaszai csak a hoszt felé haladnak
- Host-device (master-slave) kommunikáció, 1..127 device
- Egy HW, több cím: compound device
- Különböző funkciók egy címen: composite



# Kábelek, csatlakozók

- Csatlakozók:

A-type: Hosthoz közeli

B- type: device felöli

- Max. 5 m-es 2.0 kábel erei:

5V táp/föld (piros/fekete)

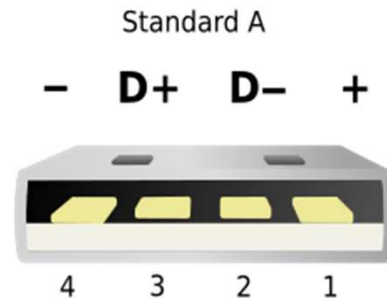
Csavart érpár adatoknak: D- /D+ (fehér/zöld)

- Max. 3 m-es 3.0 kábel

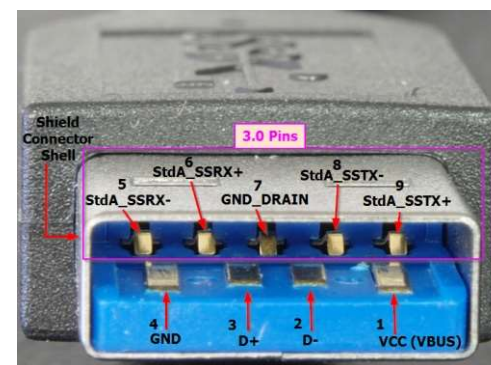
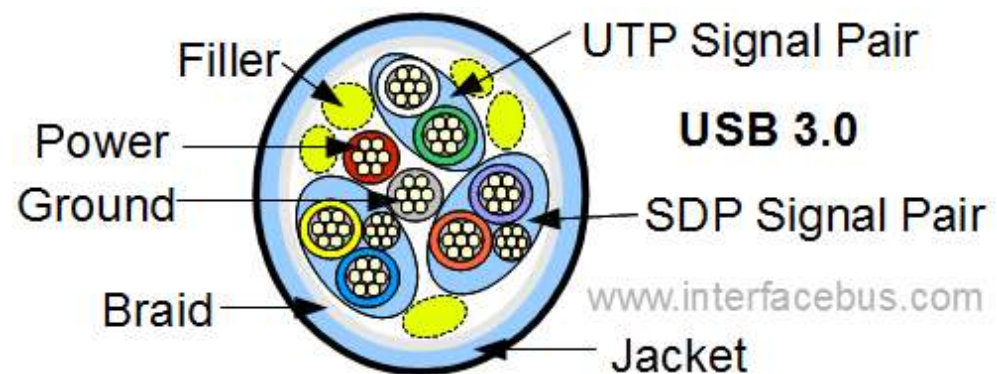
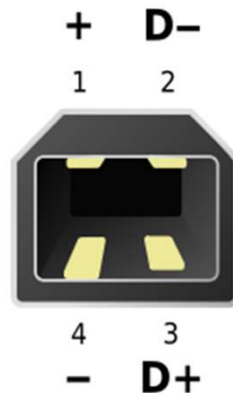
+2 pár árnyékolt kábel,

full-duplex kommunikáció

**USB**



Standard B

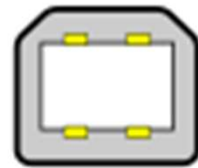


# Jelenlegi csatlakozó típusok

## USB 1.1 – 2.0



A



B



Mini-A



Mini-B

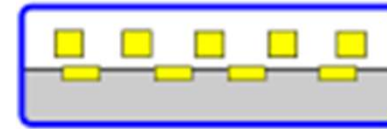


Micro-A

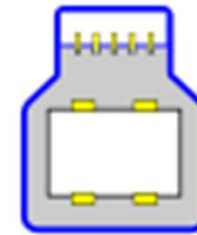


Micro-B

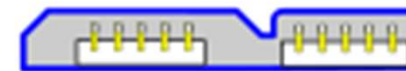
## USB 3.0



A



B



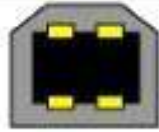
Micro-B

# Jelenlegi csatlakozó típusok

## USB CONNECOR TYPES



Type A



Type B



Type A Mini



Type B Mini



Type A Micro



Type B Mico



Type B Micro USB3



Type C

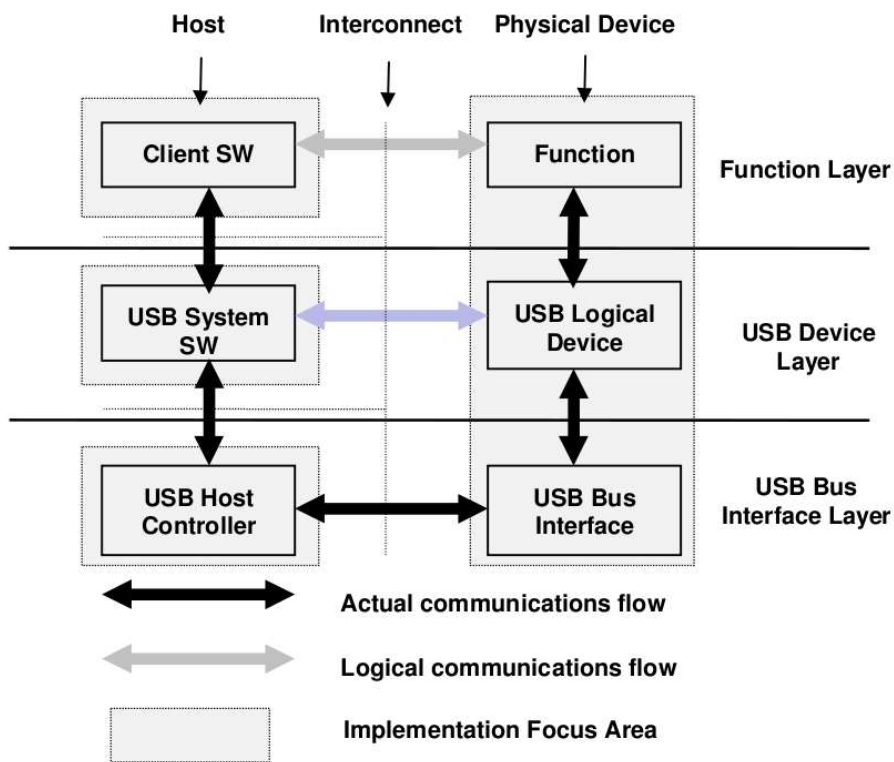


# Adatfolyam USB 2.0 → 3.0

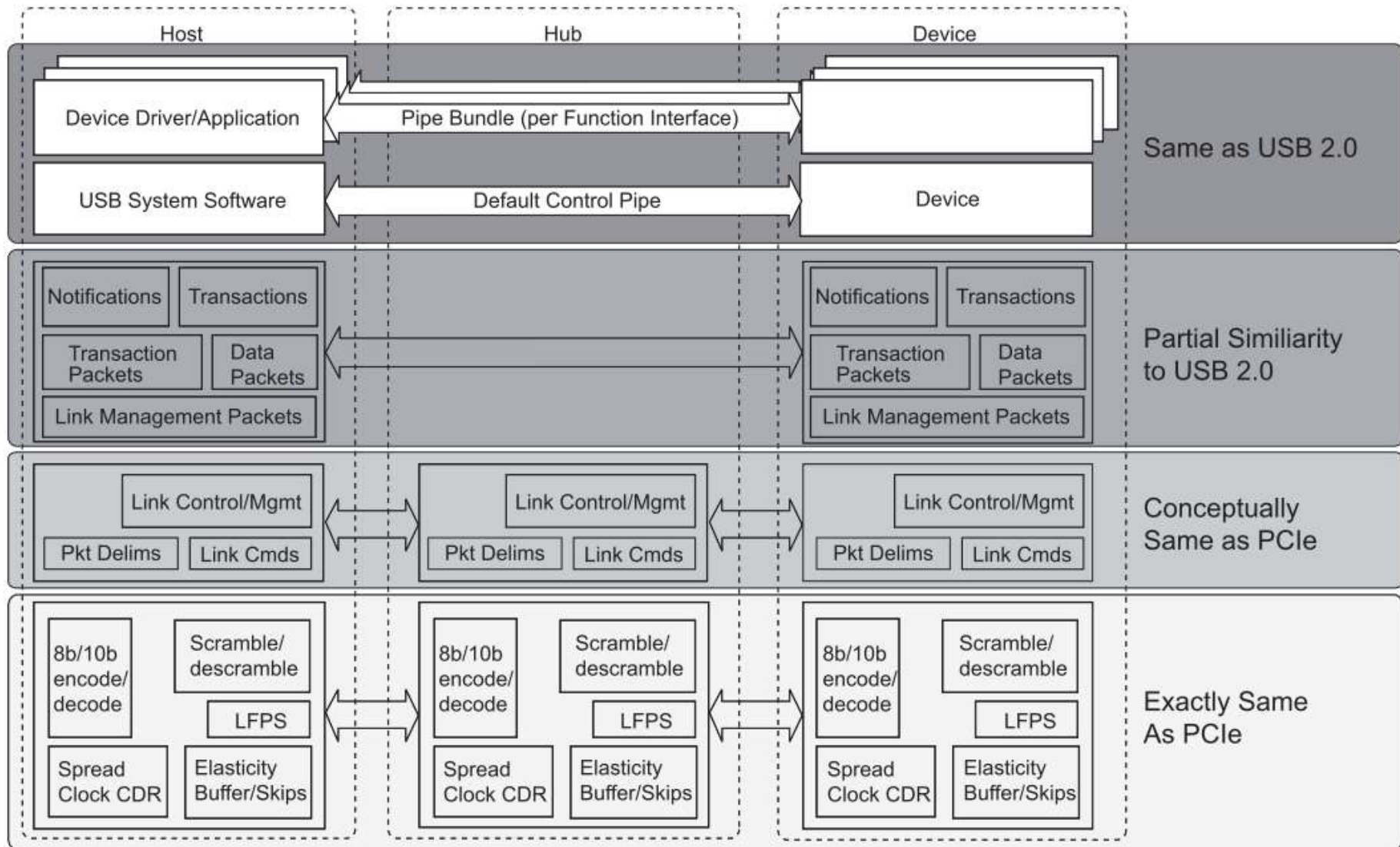
- Fizikai réteg
- Protokoll réteg
- Device/Host kommunikáció

## USB 3.0:

- Fizikai réteg PCIe-től átvéve
- Adatkapcsolati réteg (új): PCIe koncepciója szerint
- Protokoll réteg: módosított USB 2.0
- Device/Host szint: egységes



# Adatfolyam USB 3.0





# Tápellátás

## Eszköz konfigurációban megadott típus

- Low-power bus powered  
4.4-5.25V-ot elvisel, max 100 mA
- High-power bus powered  
Konfigurált állapotban max 500 mA, 4.75-5.25 V
- Self-powered

## Kikapcsolt állapot:

- low/high-powered 0.5 mA/2.5 mA
- Csak a felhúzó ellenállás elvisz 200  $\mu$ A-t!

## USB 3.0:

- Low/high powered: 150 mA/900 mA
- Battery charging mode: 1.5 A – kommunikáció nincs

# Fizikai réteg – USB 2.0

Felhúzó ellenállástól függetlenül, egységesen értelmezett busz állapotok:

	Full/High		Low		LS/FS
	D+	D-	D+	D-	
SE1	1	1	1	1	Tiltott
SE0	0	0	0	0	mint Detached
J-State	1	0	0	1	mint Idle
K-State	0	1	1	0	J ellentettje

- K a „meghajtott” állapot
- Reset jel: > 10 ms SE0 (Single Ended 0)
- EOP: End of packet: LS/FS: 2xSE0 + 1xJ, HS: szándékos bit-stuff hiba
- Suspend: >= 3 ms Idle
- Keep Alive: LS: EOP, FS: StartOfFrame packet
- WakeUp: >= 20 ms K (remote wake-up: device ébreszt)

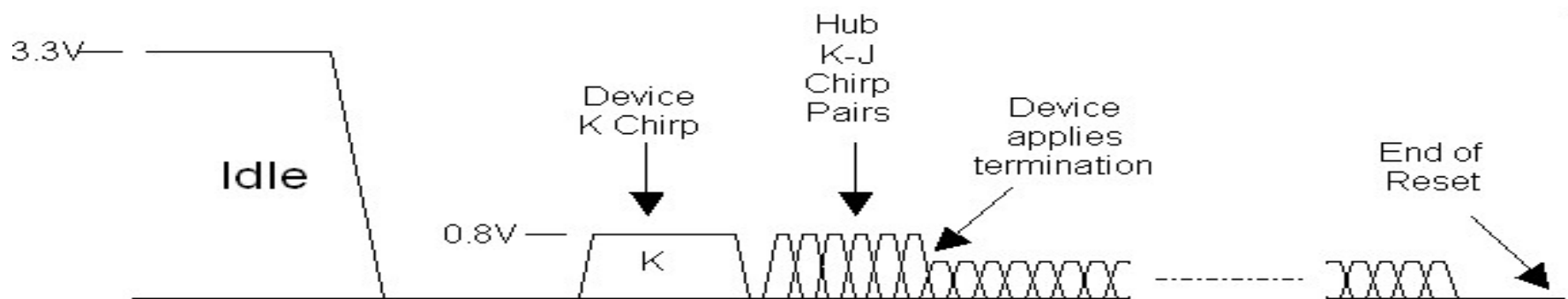
# Fizikai réteg, sebességek detektálása

A kompatibilitás miatt egyre bonyolultabb megvalósítás, protokoll

- Low/Full Speed: felhúzó ellenállás alapján
- High speed: detektálás a protokollal
- USB 3.0: vezetékszám kétszerezés

High speed detektálás:

- Idle state: SE0 mindkét végen  $\rightarrow$  45 ohm lezárás, áramgenerátoros meghajtás 17.78 mA
- Eszköz lehúzás: adatvezetéken 400  $\rightarrow$  800 mA
- High speed képesség jelzése:
  - 1, Reset=SE0, a device nem zár le, de meghajtja a buszt: K-Chirp
  - 2, A 800 mV-ra a hub KJ-Chirppel válaszol – Innen tudja a device, hogy HS hubon van



# USB On The Go

A master-slave „kiterjesztése” beágyazott eszközökhöz:

- Csatlakozás detektálása táp nélkül, kapacitás méréssel: *Attach Detection Protocol*
- Lehetőség van szerepcserére, a csatlakoztatás módja a kezdeti szerepeket dönti el (pl. fényképezőgép host/device is lehet, fordított kábelt detektálja) *Host Negotiation Protocol*
- Device is kérheti a táp bekapcsolását: *Session Request Protocol*

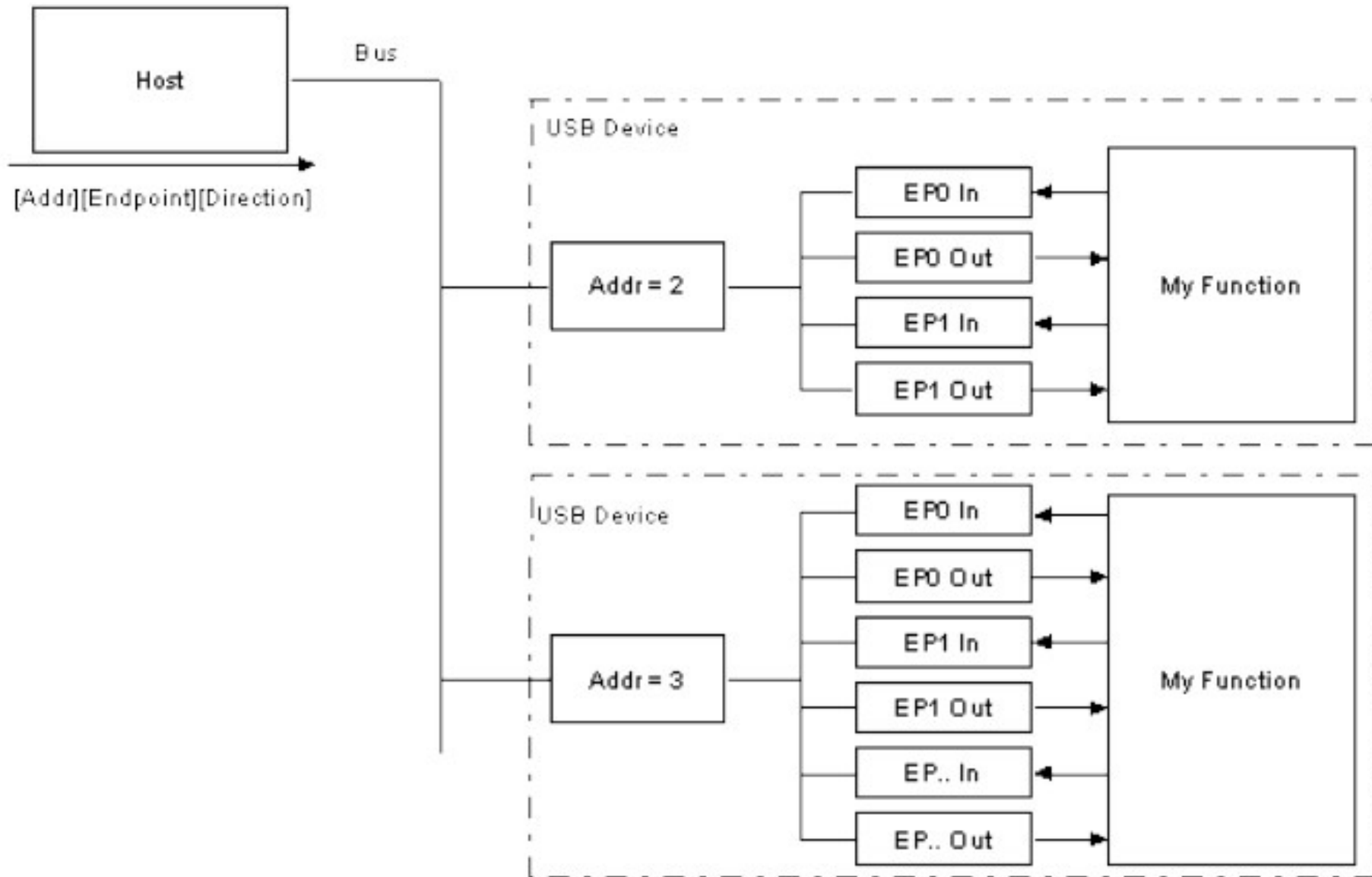
# Adatátviteli módok

- Végpontok (endpoint) közötti logikai csatorna: pipe. Max. 32 eszközönként
  - Message pipe: kétirányú, csak control transfer – a 0. végpontokon (2 pár)
  - Stream pipe: egyirányú adatátvitel egy végpontpár között
- Stream transfer típusok (Control transfer külön számít):

	Sok adat	Hibamentes	Real-time
<u>Bulk</u>	+	+	-
<u>Isochron</u>	+	-	+
<u>Interrupt</u>	-	+	+



# Endpointok és eszközök kapcsolata

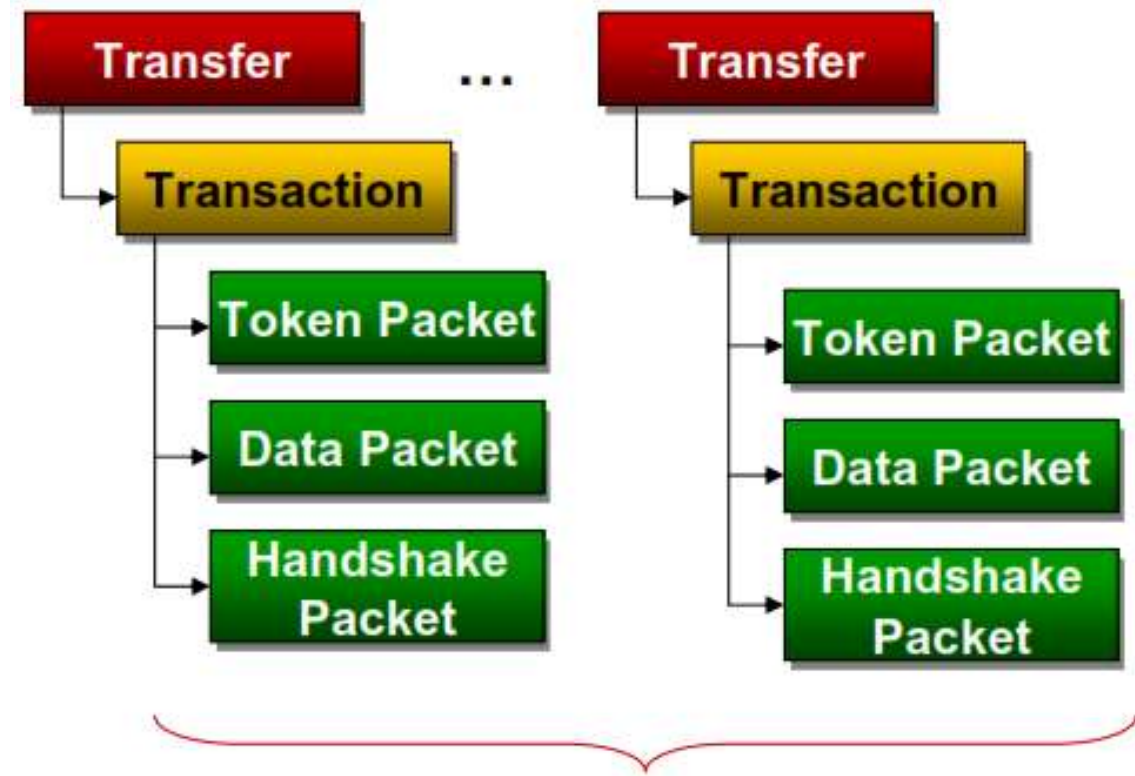


# Transzferek és tranzakciók

- A perifériákkal transzferekkel kommunikálunk
- Az LS transzfereket az LS és az FS eszközök is látják. Az FS transzfereket csak az FS eszközök látják
- Mindig a master, a root hub indítja
  - Periféria azonosítója
  - Ki- vagy bemeneti művelet
  - Átvinni kívánt adatok
    - Kimeneti irány esetén a root hub teszi a buszra
    - Bemeneti irány esetén a megcélzott periféria
- A tranzakciók transzfereket, a transzferek kereteket alkotnak
- Minden keret 1 ms ideig tart (FS sebességgel: 1500 bájt/keret)
- Minden keret több transfert is szállíthat


# Tranzakciók

- Az USB Transzfer Tranzakciókra van bontva:
- A tranzakciók több csomagból állnak



- Transfers are divided into transactions.
- Transactions are made up of packets.
- The host controls transfers by allocating transactions to a frame.
- Transfers may span multiple frames.

# Adatátvitel

- A transzfer tranzakciók sorozatából épül fel
- Egy tranzakciót csomagokból (packet) épül fel:
  - Token: „fejléc”, megadja a tranzakció típusát
  - Data: opcionális adat
  - Handshake: státusz információ, isochron átvitelnél nincs
- Egy csomag részei: 
  - SYNC órajel szinkronizáció (8/32 bit LS-FS/HS-re)
  - PID: Packet ID 4 bit, negáltan és ponáltan is átvive
  - DATA/CRC: PID függő tartalom
  - EOP: end of packet

# Csomagok

PID Type	PID Name	PID<3:0>*
Token	OUT	0001b
	IN	1001b
	SOF	0101b
	SETUP	1101b
Data	DATA0	0011b
	DATA1	1011b
	DATA2	0111b
	MDATA	1111b
Handshake	ACK	0010b
	NAK	1010b
	STALL	1110b
	NYET	0110b
Special	PRE	1100b
	ERR	1100b
	SPLIT	1000b
	PING	0100b
	Reserved	0000b

- Az átvitelt hardver támogatja, fontos a PID gyors dekódolása  
SIE: Serial Interface Engine
- LSB first átvitel, a PID alsó 2 bitje azonosítja a csomag típusát

# TOKEN csomagok

- IN,OUT: kommunikáció iránya a hub szemszögéből
- SETUP: Control transfer

Sync	PID	ADDR	ENDP	CRC5	EOP
	8 bits	7 bits	4 bits	5 bits	

A 0. cím eszköz csatlakoztatáshoz fenntartott, a 0. EP SETUP csomagnak

- SOF: Start of Frame:

Sync	PID	Frame No.	CRC5	EOP
	8 bits	11 bits	5 bits	

LS: KeepAlive = EOP signal van csak helyette

FS: 1 ms időkeret, pl. isochron tranfer egy időegysége

HS: 125 us-os mikroframe határa

# DATA csomagok

- DATA0/1: LS/FS busznál alternálva használt
- DATA2, MDATA: csak HS isochron transfernél
  - DATAx IN transfernél, x a hátralévő tranzakciók száma a mikroframe-ben
  - OUT transfernél az utolsó transfer DATAx típusú, a többi MDATA

Sync	PID	DATA	CRC16	EOP
	8 bits	(0-1024) x 8 bits	16 bits	

# Transzfer, tranzakció

Control transzfer:

- SETUP fázis: SETUP → DATA0 (8byte) → ACK
- DATA fázis:  $k \cdot (\text{IN/OUT} \rightarrow \text{DATA1/0} \rightarrow \text{ACK})$
- STATUS fázis: OUT/IN → ACK

Bulk /Interrupt:  $k \cdot (\text{IN/OUT} \rightarrow \text{DATA0/1} \rightarrow \text{ACK})$

Isochron:  $k \cdot (\text{IN/OUT} \rightarrow \text{DATA0/1} \rightarrow \text{ACK})$

Payload méret:

	LS	FS	HS
Control	8	64	64
Bulk	--	64	512
Isochron	--	1023	1024
Interrupt	8	64	1024



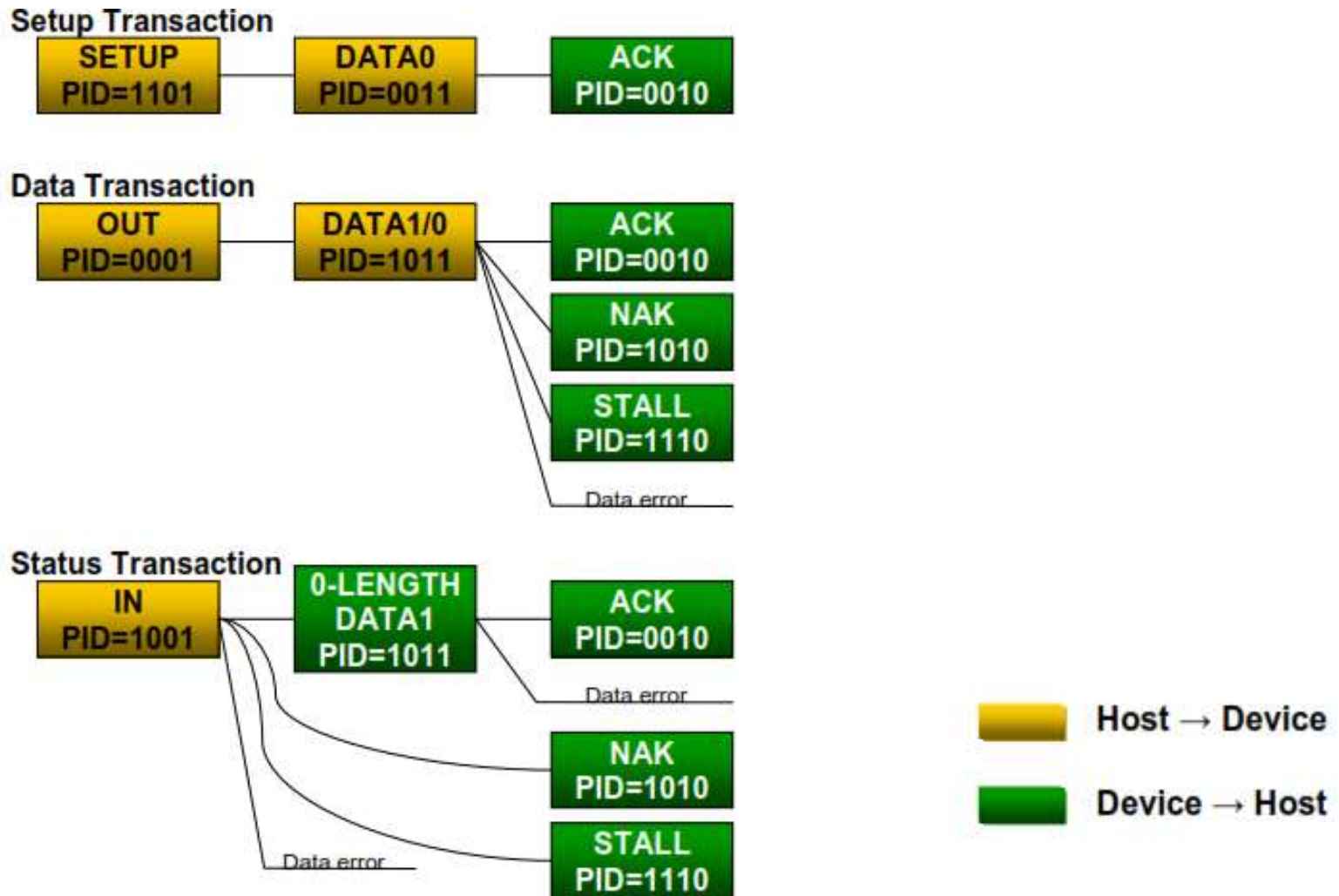
# Transzfer típusok

Transfer Type	Stages (Transactions)	Phases (Packets)	Comments
Control	Setup	Token	<ul style="list-style-type: none"> <li>◆ Enables host to read configuration information, set addresses and select configurations</li> <li>◆ Only transfer that is required to be supported by peripherals</li> <li>◆ Has both IN and OUT transfers to a single endpoint</li> </ul>
		Data	
		Handshake	
	Data (IN or OUT) (optional)	Token	
		Data	
		Handshake	
	Status (IN or OUT)	Token	
		Data	
		Handshake	
Bulk	Data (IN or OUT)	Token	<ul style="list-style-type: none"> <li>◆ Non-critical data transfers</li> <li>◆ Bandwidth allocated to the host</li> <li>◆ Good for file transfer where time critical data is not required</li> </ul>
		Data	
		Handshake	
Interrupt	Data (IN or OUT)	Token	<ul style="list-style-type: none"> <li>◆ Periodic transfers on the time base conveyed during enumeration</li> <li>◆ Host guarantees attention before this elapsed time</li> </ul>
		Data	
		Handshake	
Isochronous	Data (IN or OUT)	Token	<ul style="list-style-type: none"> <li>◆ Guaranteed delivery time of packets for data streaming</li> <li>◆ No-retransmitting of data allowed</li> </ul>
		Data	

# Control transzfer

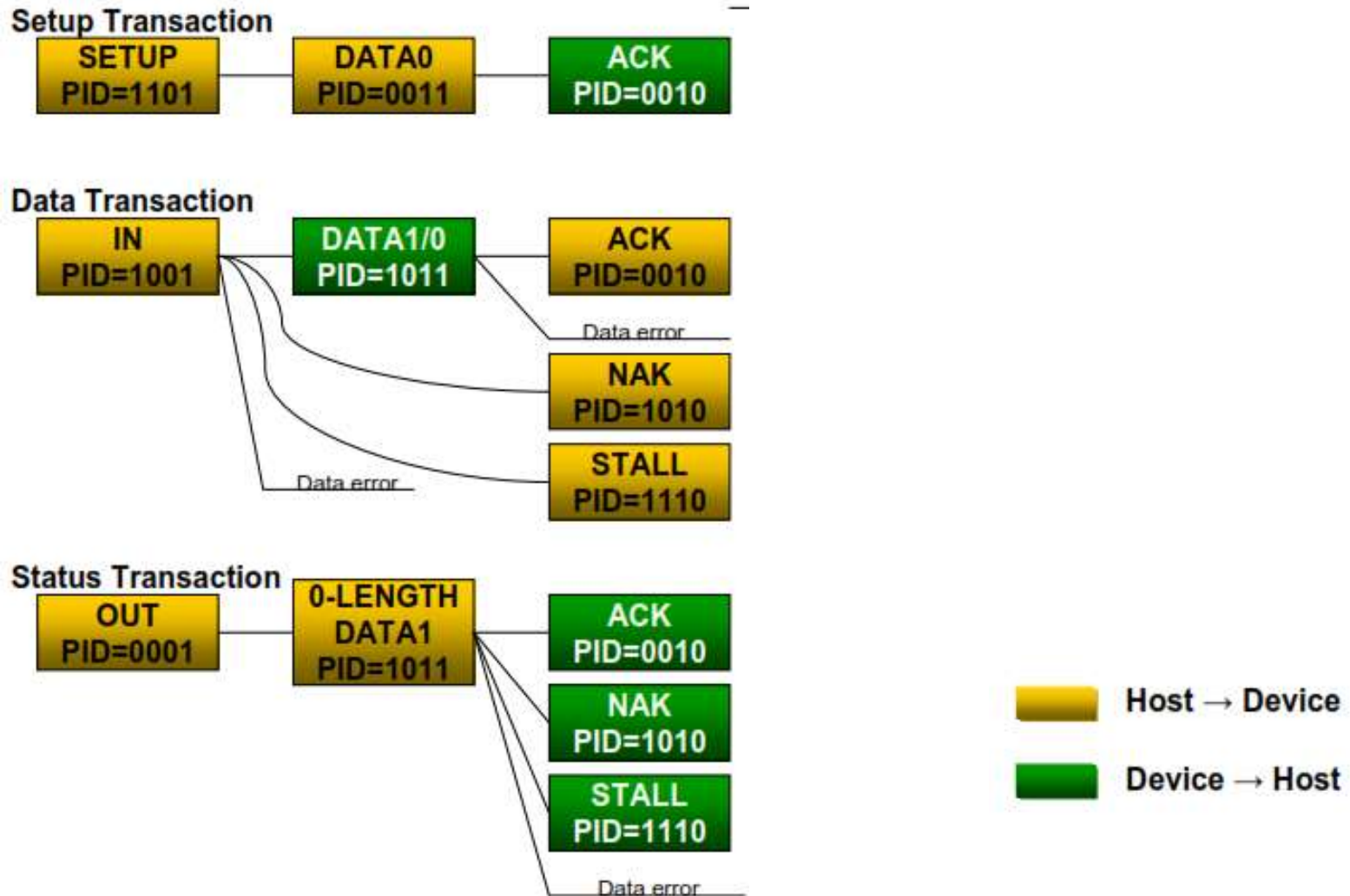
# Control OUT transzfer példa

- Control OUT Transfer: Setup, Data, Handshake



# Control IN transzfer példa

- Control IN Transzfer példa: Setup, Data, Handshake



# Setup tranzakció felépítése

- A Setup tranzakcióban átvitt mezők
  - **bmRequestType:** Az átvitel irányáról információ, a kérés típusa, és a kérés célja (standard, class, vendor, reserved)
  - **bRequest:** A tényleges kérés azonosítója
  - **wValue:** A kéréssel kapcsolatos információ amit a host a device-nak eljuttat. Pl. a Set\_Address kérésnél az eszköt új címe
  - **wIndex:** szintén a host által a device-nak adott információ. Pl. például egy endpoint azonosítása. Endpoint, bit 3-0 az endpoint azonosítója, bit 7 az endpoint irány azonosítója (0 OUT, 1 IN). Interface esetében, bit 7-0 az interfész azonosítót adja meg.
  - **wLength:** Az adat fázisban elküldött byte-ok száma. A host -> device irányban a tényleg byte szám, másik irányban a maximális byte szám amit a device küldhet. Az értéke sokszor 0.

# Setup tranzakció felépítése

Offset	Field	Size	Value	Description
0	bmRequestType	1	Bitmap	D7 Data direction 0 – Host to Device 1 – Device to Host
				D6:5 Type 00 – Standard 01 – Class 10 – Vendor 11 – Reserved
				D4:0 Recipient 00000 – Device 00001 – Interface 00010 – Endpoint 00011 – Other Else Reserved
1	bRequest	1	Value	Specific Request
2	wValue	2	Value	According to request
4	wIndex	2	Index or Offset	According to request
6	wLength	2	Count	Number of bytes to be transferred in data stage if there is any

# Standard USB kérések (bRequest)

## ■ GetStatus

- Státusz információ lekérdezése a device-től, interfésztől, vagy endpoint-tól
- bmRequestType azonosítja, hogy device-ről interfésztől, vagy endpoint-ról van szó
- wIndex azonosítja az endpoint-ot, vagy interfészt, ha nem a device státuszt akarjuk lekérdezni

# Standard USB kérések (bRequest)

## ■ GetStatus

### ○ Device status

Device Status		
Status Bit	Purpose	Comment
D0	Self Powered	1 : Self Powered 0 : Bus Powered
D1	Remote Wakeup	1 : Enable 0 : Disable
D2-D15	Reserved	Must be set to 0

### ○ Interface status

Interface Status		
Status Bit	Purpose	Comment
D0-D15	Reserved	Must be set to 0

### ○ Endpoint status

Endpoint Status		
Status Bit	Purpose	Comment
D0	Halt	1 : Endpoint Halted
D1-D15	Reserved	Must be set to 0



# Standard USB kérések (bRequest) nem teljes lista

## ■ SET\_ADDRESS

- Új cím adása a device-nak (1-127.). Minden device 0-ás címmel indul, és a host az inicializálás során ad neki ezzel a paranccsal tényleges címet.
- wValue: a device új címe

# Standard USB kérések (bRequest) nem teljes lista

- GET\_DESCRIPTOR
  - Különféle leírók lekérdezése

Descriptor Type	wValue(Higher Byte)
Device	0x01
Configuration	0x02
String	0x03
Interface	0x04
Endpoint	0x05
Device Qualifier	0x06
Other Speed Configuration	0x07
Interface Power	0x08
On The Go(OTG)	0x09

# Standard USB kérések (bRequest) nem teljes lista

- SET\_FEATURE/CLEAR\_FEATURE
  - A device, interfész, vagy endpoint bizonyos tulajdonságainak be, vagy kikapcsolása
  - bmRequestType azonosítja, hogy device-ről interfésztől, vagy endpoint-ról van szó
  - wIndex azonosítja az endpoint-ot, vagy interfészt, ha nem a device státuszt akarjuk lekérdezni
  - wValue azonosítja a tulajdonságot (pl. Endpoint halt egy endpointra)

# Standard USB kérések (bRequest) nem teljes lista

- SET\_DESCRIPTOR
  - Leírók frissítése
- SET\_CONFIGURATION
  - A wValue adja meg, hogy melyik konfiguráció lesz aktív, mindig csak egy konfiguráció lehet aktív egyszerre. A legtöbb eszköznek nincs is több konfigurációja
- GET\_CONFIGURATION
  - Aktív konfiguráció lekérdezése
- SET\_INTERFACE
- GET\_INTERFACE
- SYNCH\_FRAME

# Egy tényleges Control transzfer IN példa

Transfer  
Transaction  
Packet  
Hierarchy

Transfer	F	Control	ADDR	ENDP	bRequest	windex	Port Status bits	Port Change bits	Time			
4	S	GET	2	0	GET_STATUS	For Port # 1	0000000100000000	0000000000000000	0 ns			
<b>Transaction 20</b>												
Transaction	F	SETUP	ADDR	ENDP	D	T	R	bRequest	wValue	windex	wLength	ACK
20	S	0xB4	2	0	D->H	C	O	GET_STATUS	0x0000	For Port # 1	4	0x4B
<b>Packet # 444</b>												
Packet #	F	Sync	SETUP	ADDR	ENDP	CRC5	EOP	Idle				
444	S	00000001	0xB4	2	0	0x15	3.00	2				
<b>Packet # 445</b>												
Packet #	F	Sync	DATA0	DATA	CRC16	EOP	Idle					
445	S	00000001	0xC3	A3 00 00 00 01 00 04 00	0x6FA5	2.50	3					
<b>Packet # 446</b>												
Packet #	F	Sync	ACK	EOP	Idle							
446	S	00000001	0x4B	3.00	11785							
<b>Transaction 21</b>												
Transaction	F	IN	ADDR	ENDP	T	DATA	ACK					
21	S	0x96	2	0	1	00 01 00 00	0x4B					
<b>Packet # 448</b>												
Packet #	F	Sync	IN	ADDR	ENDP	CRC5	EOP	Idle				
448	S	00000001	0x96	2	0	0x15	3.00	3				
<b>Packet # 449</b>												
Packet #	F	Sync	DATA1	DATA	CRC16	EOP	Idle					
449	S	00000001	0xD2	00 01 00 00	0x75D8	2.50	7					
<b>Packet # 450</b>												
Packet #	F	Sync	ACK	EOP	Idle							
450	S	00000001	0x4B	2.50	11814							
<b>Transaction 22</b>												
Transaction	F	OUT	ADDR	ENDP	T	DATA	ACK					
22	S	0x87	2	0	1		0x4B					
<b>Packet # 452</b>												
Packet #	F	Sync	OUT	ADDR	ENDP	CRC5	EOP	Idle				
452	S	00000001	0x87	2	0	0x15	2.50	2				
<b>Packet # 453</b>												
Packet #	F	Sync	DATA1	DATA	CRC16	EOP	Idle					
453	S	00000001	0xD2		0x0000	3.00	2					
<b>Packet # 454</b>												
Packet #	F	Sync	ACK	EOP	Idle							
454	S	00000001	0x4B	3.00	11849							

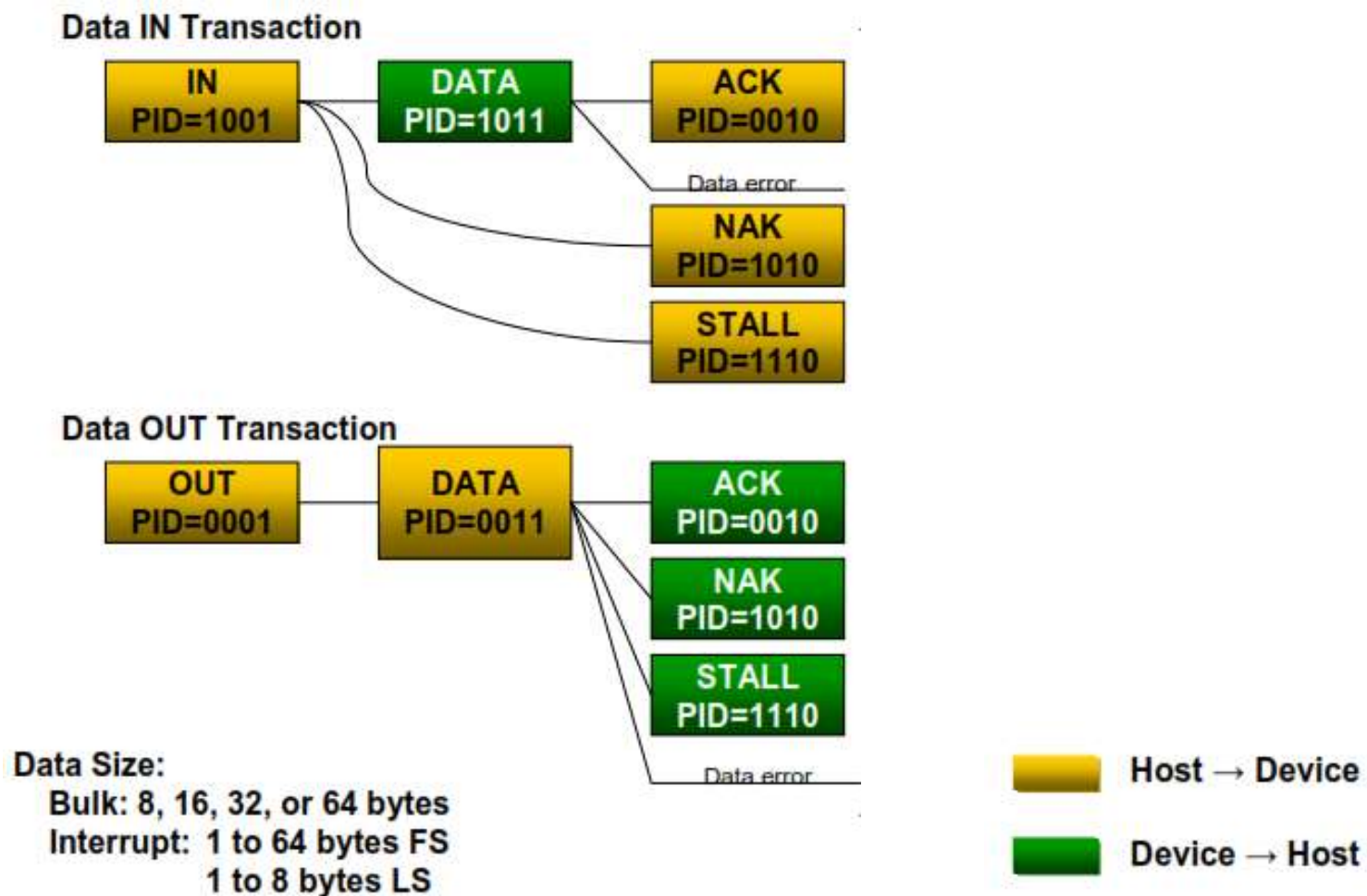
SETUP Phase

DATA Phase

Status Phase  
0 length field

# Példa transzferekre

- Bulk és Interrupt Transfer példa: IN/OUT, Data, Handshake

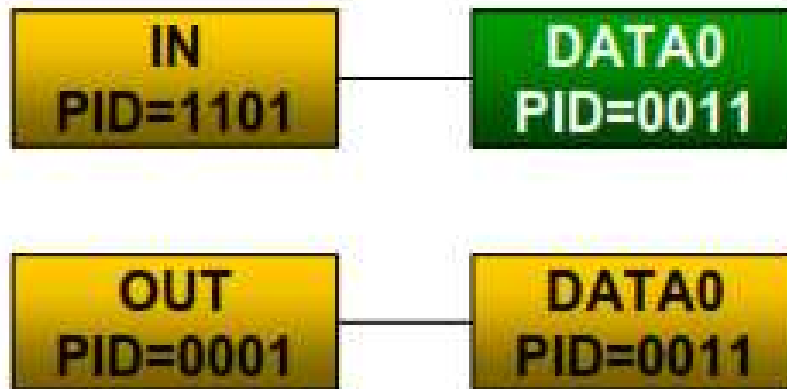


# Tényleges Interrupt transzfer



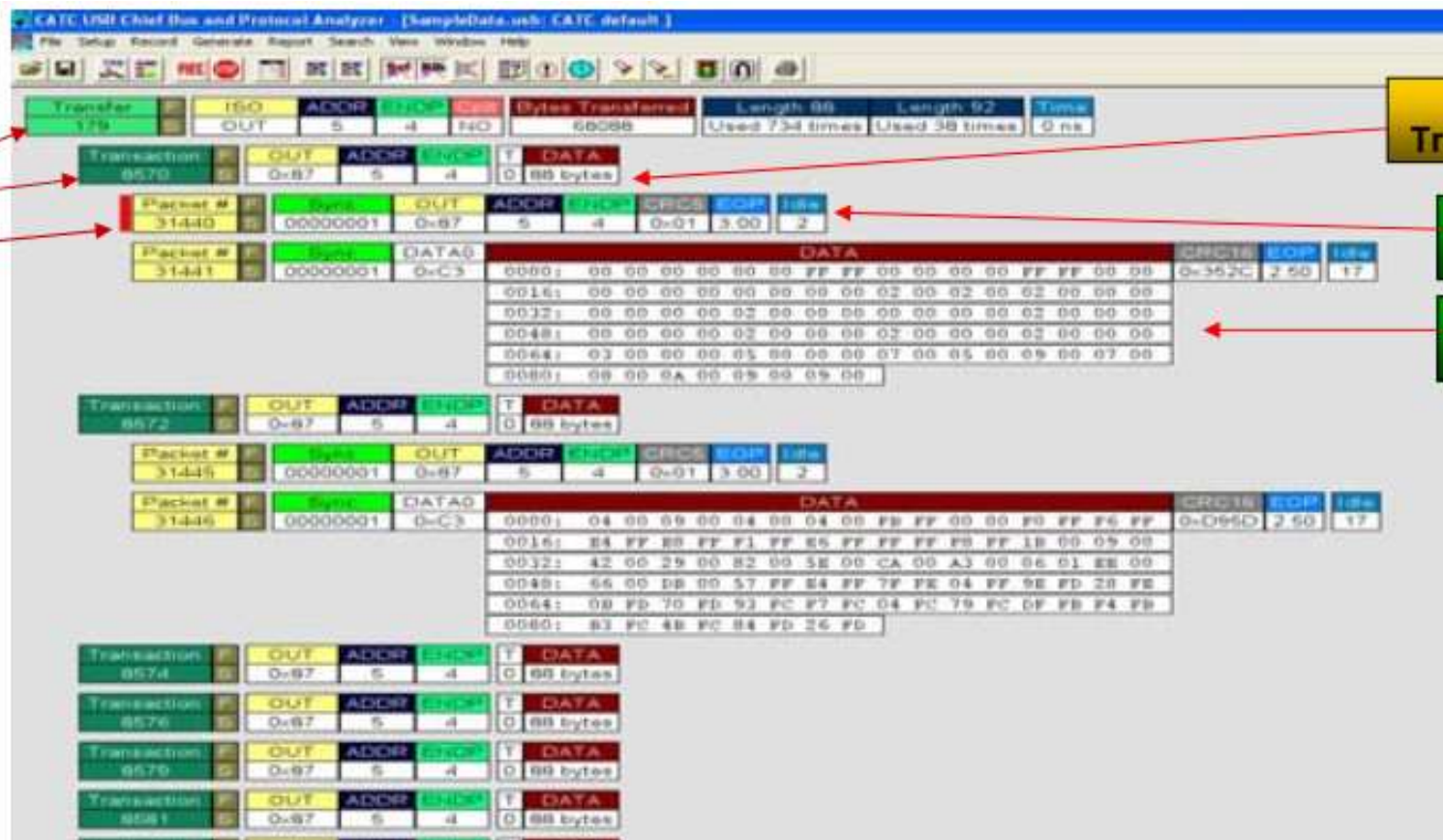
# Példa transzferekre

- Isochronon átvitel: IN/OUT, Data





# Tényleges Isochron átvitel



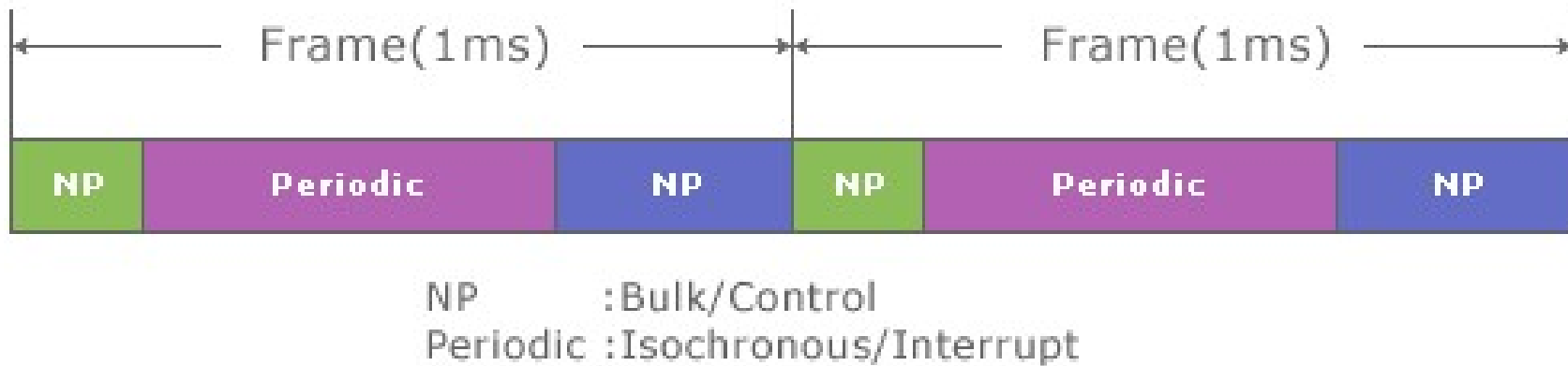


# Keretek

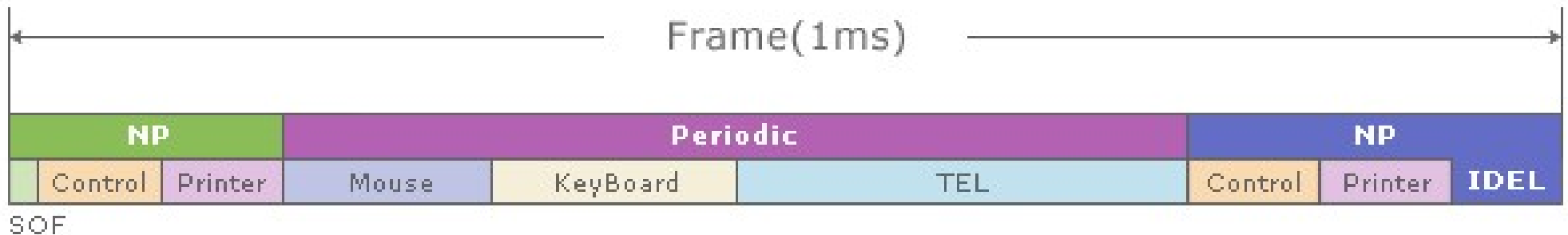
- Az izokron és interrupt tranzakcióknak elsőbbsége van
  - A keret max. 90%-a
- Ha új periféria csatlakozna, amivel több lenne ez 90%-nál, akkor nem engedik belépni
- A fennmaradó 10%-ban elsőbbsége van a control tranzakcióknak
- Maradék sávszélesség: Bulk
  
- Low Speed, Full Speed: 1ms-os keretek, High Speed 125us –os micro keretek (egy keret 8 mikro keretből áll)

# Keretek

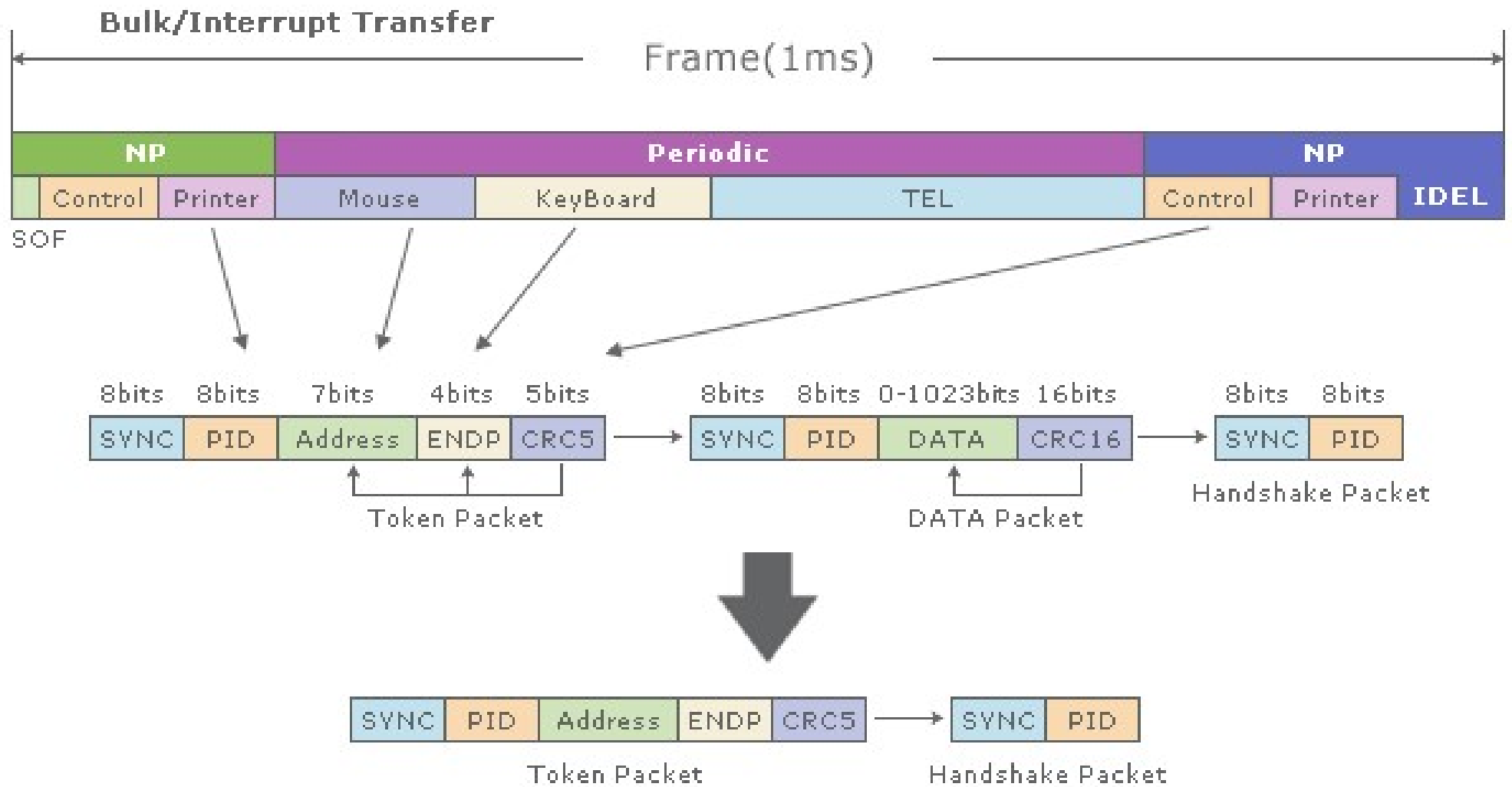
## ■ Példa keretekre



## ■ Példa egy keretre



# Részletesebb példa

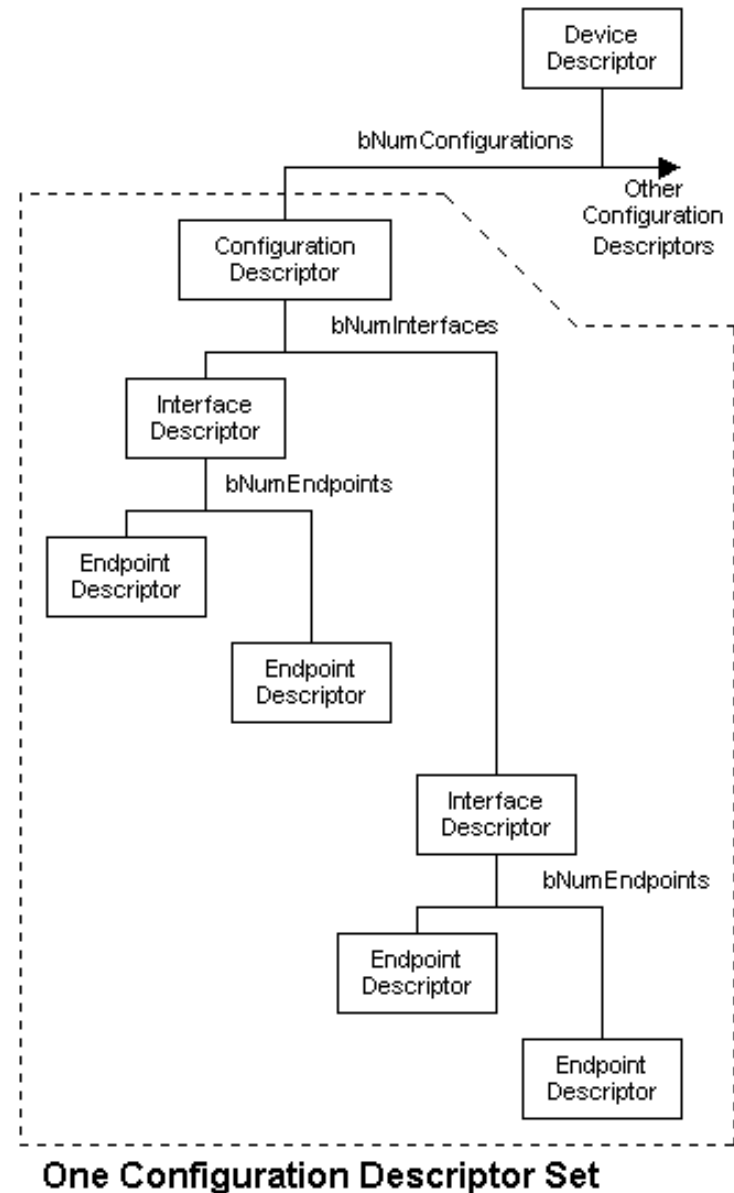


# Új eszköz csatlakoztatása

- Felhúzó ellenállás, K-Chirp → eszköz detektálva
- Reset (egyszerre csak egy eszközre!), az eszköz a 0 címre „hallgat”
- 0. cím 0. EP GetDeviceDescriptor: maximális csomagméret lekérdezése
- Reset, SetAddress
- Device, Configuration és String descriptor lekérése
- Driver betöltése
- SetConfiguration

# Leírók

- Device: egyedi
- Configuration: egyszerre csak 1 aktív (ritkán van több)
- Interface: composite eszközöknél párhuzamosan aktívak (pl. VOIP telefon, fejlesztő kártya board controller-e)



# Device descriptor

- Minden eszköznek egy van belőle
- Főbb információi
  - USB revision number
  - Product ID
  - Vendor ID
  - A konfigurációs leírások száma
- A driverek betöltése a Product ID, Vendor ID páros alapján történik meg



# Device descriptor

Offset	Field	Size	Value	Description
0	bLength	1	Number	Size of the Descriptor in Bytes (18)
1	bDescriptorType	1	Constant	Device Descriptor (0x01)
2	bcdUSB	2	BCD	USB Specification Number which device complies too.
4	bDeviceClass	1	Class	Class Code (by USB Org) If equal to Zero, each interface specifies it's own class code If equal to 0xFF, the class code is vendor specified. Otherwise field is valid Class Code.
5	bDeviceSubClass	1	SubClass	Subclass Code (by USB Org)
6	bDeviceProtocol	1	Protocol	Protocol Code (by USB Org)
7	bMaxPacketSize	1	Number	Maximum Packet Size for Zero Endpoint. Valid Sizes are 8, 16, 32, 64
8	idVendor	2	ID	Vendor ID (by USB Org)
10	idProduct	2	ID	Product ID (by Manufacturer)
12	bcdDevice	2	BCD	Device Release Number
14	iManufacturer	1	Index	Index of Manufacturer String Descriptor
15	iProduct	1	Index	Index of Product String Descriptor
16	iSerialNumber	1	Index	Index of Serial Number String Descriptor
17	bNumConfigurations	1	Integer	Number of Possible Configurations

```

const uint8_t DeviceDescriptor[SIZ_DEVICE_DESC] =
{
    SIZ_DEVICE_DESC, /* bLength */
    0x01, /* bDescriptorType */
    0x00, /* bcdUSB, version 2.00 */
    0x02,
    0x00, /* bDeviceClass : each interface define the device class */
    0x00, /* bDeviceSubClass */
    0x00, /* bDeviceProtocol */
    0x40, /* bMaxPacketSize0 0x40 = 64 */
    0x83, /* idVendor (0483) */
    0x04,
    0x21, /* idProduct */
    0x57,
    0x00, /* bcdDevice 2.00*/
    0x02,
    1, /* index of string Manufacturer */
    /**/
    2, /* index of string descriptor of product*/
    /**/
    3, /* */
    /** */
    /** */
    0x01 /*bNumConfigurations */
};
    
```

# Device class codes

Base Class	Descriptor Usage	Description
00h	Device	<a href="#">Use class information in the Interface Descriptors</a>
01h	Interface	<a href="#">Audio</a>
02h	Both	<a href="#">Communications and CDC Control</a>
03h	Interface	<a href="#">HID (Human Interface Device)</a>
05h	Interface	<a href="#">Physical</a>
06h	Interface	<a href="#">Image</a>
07h	Interface	<a href="#">Printer</a>
08h	Interface	<a href="#">Mass Storage</a>
09h	Device	<a href="#">Hub</a>
0Ah	Interface	<a href="#">CDC-Data</a>
0Bh	Interface	<a href="#">Smart Card</a>
0Dh	Interface	<a href="#">Content Security</a>
0Eh	Interface	<a href="#">Video</a>
0Fh	Interface	<a href="#">Personal Healthcare</a>
10h	Interface	<a href="#">Audio/Video Devices</a>
DCh	Both	<a href="#">Diagnostic Device</a>
E0h	Interface	<a href="#">Wireless Controller</a>
EFh	Both	<a href="#">Miscellaneous</a>
FEh	Interface	<a href="#">Application Specific</a>
FFh	Both	<a href="#">Vendor Specific</a>

# Configuration descriptor

- Milyen interfészei vannak az adott eszköznek
- Mennyi energiát fogyaszt
- Egyszerre egy konfiguráció lehet aktív, de a legtöbb eszköznek csak egy konfigurációja van
  
- Az inicializálás során a device descriptor kiolvasása után dönti el a host, hogy melyik konfigurációt fogja használni

Field	Value	Meaning
bLength	9	Valid length
bDescriptorType	2	CONFIGURATION
wTotalLength	34	Total combined size of this set of descriptors
bNumInterfaces	1	Number of interfaces supported by this configuration
bConfigurationValue	1	Value to use as an argument to the SetConfiguration() request to select this configuration
iConfiguration	0	Index of string descriptor describing this configuration
bmAttributes (Self-Powered)	0	Bus-Powered
bmAttributes (Remote Wakeup)	1	Yes
bmAttributes (Other bits)	0x80	Valid
bMaxPower	100 mA	Maximum Current Drawn by Device in This Configuration

# Interface descriptor

- Az interfész az endpointok csoportjának funkciók szerinti rendezése, amely egy funkcionalitást lát el
- Például vannak több funkciós eszközök, ilyenkor mindegyik interfész leíró egy funkcionalitást ad meg
- Pl. multifunkciós Fax/scanner/printer
  - Interface 1: fax funkciók
  - Interface 2: Scanner funkció
  - Interface 3: printer funkció
- Egyszerre több interfész is lehet aktív

Field	Value	Meaning
bLength	9	Valid length
bDescriptorType	4	INTERFACE
bInterfaceNumber	0	Zero-based Number of this Interface.
bAlternateSetting	0	Value used to select this alternative setting for the interface identified in the prior field
bNumEndpoints	1	Number of endpoints used by this interface (excluding endpoint zero).
bInterfaceClass	0x03	HID
bInterfaceSubClass	0x01	Boot Interface
bInterfaceProtocol	0x02	Mouse
iInterface	0	Index of string descriptor describing this Interface

# Endpoint descriptor

- Az endpoint
  - típusa
  - Íránya
  - Polling intervalum
  - Max packet size
- Az endpoint 0 mindig létezik és mindig control endpoint

Field	Value	Meaning
bLength	7	Valid length
bDescriptorType	5	ENDPOINT
bEndpointAddress	0x81	Endpoint 1 - IN
bmAttributes	0x03	Interrupt. Data Endpoint.
wMaxPacketSize	0x0004	Maximum Packet Size is 4
bInterval	0x0A	10 Frames (10 ms)

# Class specifikus leírók

- A legtöbb Class rendelkezik saját leírókkal
- Ezek a Class-on belüli tulajdonságokat, a kommunikációs protokoll részleteit stb. írják le.

Field	Value	Meaning
bLength	9	Valid length
bDescriptorType	0x21	HID
bcdHID	0x0110	HID Class Spec Version
bCountryCode	0	Not Supported
bNumDescriptors	1	Number of Descriptors
bDescriptorType	REPORT	Descriptor Type 34
wDescriptorLength	52	Descriptor Length

# USB beágyazott rendszerekben

## ■ FTDI

- FT2xxx : Soros-USB konverter (FS,HS)
- Vinculum: device szintű megoldás (FS)

## ■ Native USB

- A kontroller gyártója általában Firmware library-t is ad
- „Fizetős” megoldások

## ■ PC oldal:

- Szabványos eszköz esetén nincs szükség külön driverre
- Egyedi eszköz:
  - libusb32 driver segítségével
  - LabVIEW: VISA Driver wizard

# Hasznos linkek

- <http://www.usb.org/developers/>
- [http://en.wikipedia.org/wiki/Universal\\_Serial\\_Bus](http://en.wikipedia.org/wiki/Universal_Serial_Bus)
- <https://www.usbmadesimple.co.uk/>
- <http://www.beyondlogic.org/usbnutshell/usb1.shtml>
- <http://www.ftdichip.com>
- <http://sourceforge.net/apps/trac/libusb-win32/wiki>