

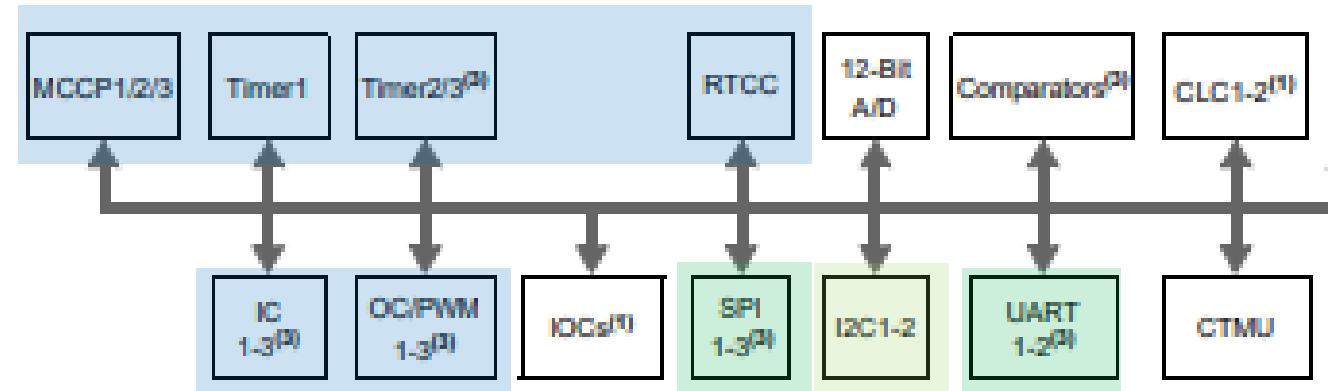
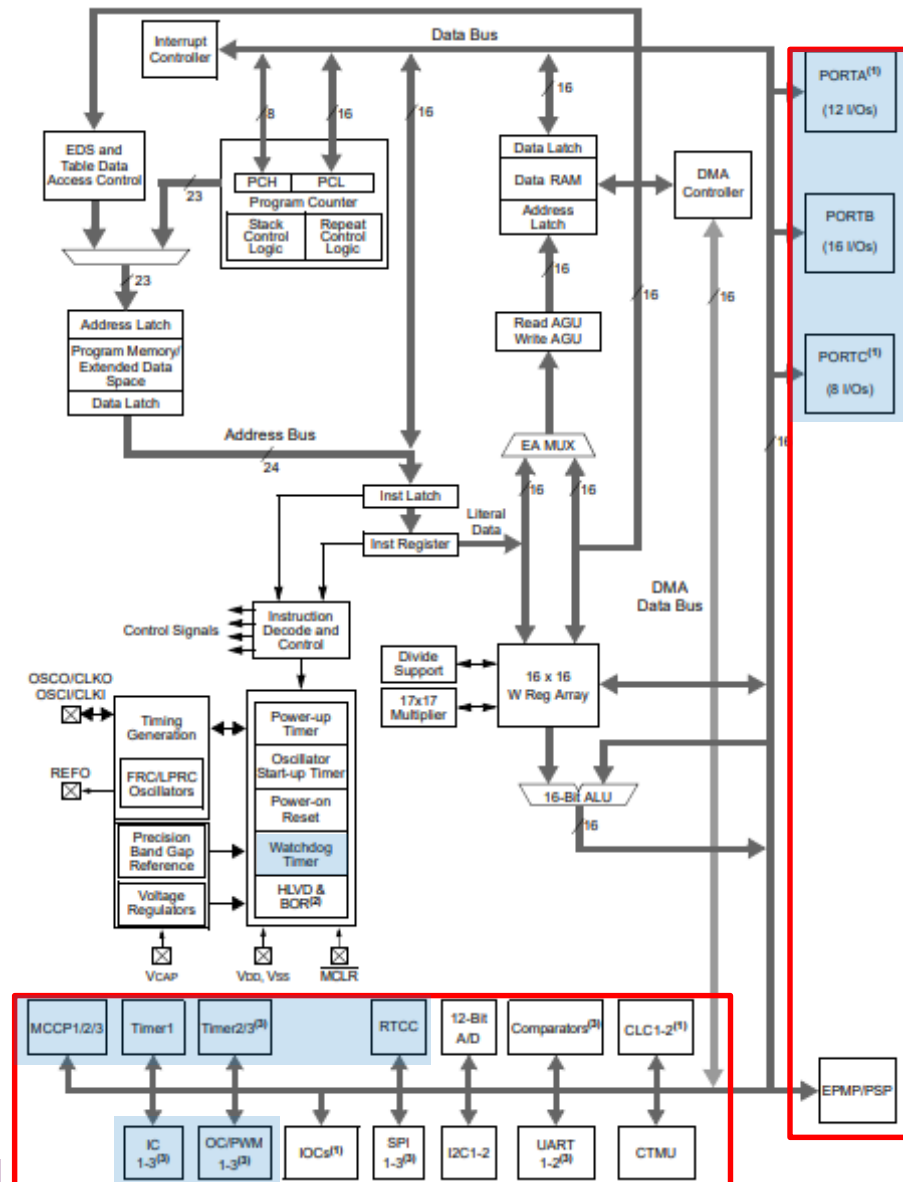
# *Digitális technika 2.*

*BMEV8IAA06*

## *9. előadás*

*Kommunikációs perifériák*

# Perifériák a PIC24FJ256GA705-ben



# Láttuk eddig:

- Gyakran egy mikrokontrollerre írt szoftvernek ismétlődő, azonos jellegű feladatokat kell végezzen
  - Pl. időzítés, várakozás, négyszögjel előállítás, adattovábbítás
  - Ezeket nem mindig hatékony szoftveresen elvégezni
- A perifériák valamilyen speciális, gyakran ismétlődő feladat megoldására készült hardver egységek a processzor mellett.
- A perifériák a vezérlő regisztereikbe írt számokkal vezérelhetők
- Láttuk, hogy nagyon sok periféria van
  - Ezek egy része időzítő alapú
    - ezeket láttuk a múlt órán
  - Másik részük pedig kommunikációval kapcsolatos
    - erről tanulunk most

- Egyszerre átvitt bitek száma szerint:
  - Párhuzamos adattovábbítás
    - Egymás mellett sok „párhuzamos” adatvezeték
    - Néhány vezérlőjel, ami jelzi az adat érvényességét, címet, irányt, stb.
    - Már láttuk, a sín is tulajdonképpen ilyen
  - Bitsoros átvitel
    - Egyszerre egy (vagy kevés) adatvezeték
    - Néhány vezérlő jel (jellemzően órajel, esetleg irány, kiválasztójel)
    - Már láttuk, amikor a 2. laboron kezeltünk kijelzőt, az is egy bitsoros „protokoll” szerint kommunikál.
- Adatátvitel iránya szerint:
  - Egyirányú átvitel (simplex)
  - Kétirányú (duplex)
    - Halfduplex (egyszerre egyirányú, de az irány forgatható)
    - Fullduplex (folyamatosan, egy időben is kétirányú)

# Bitsoros adattovábbítás

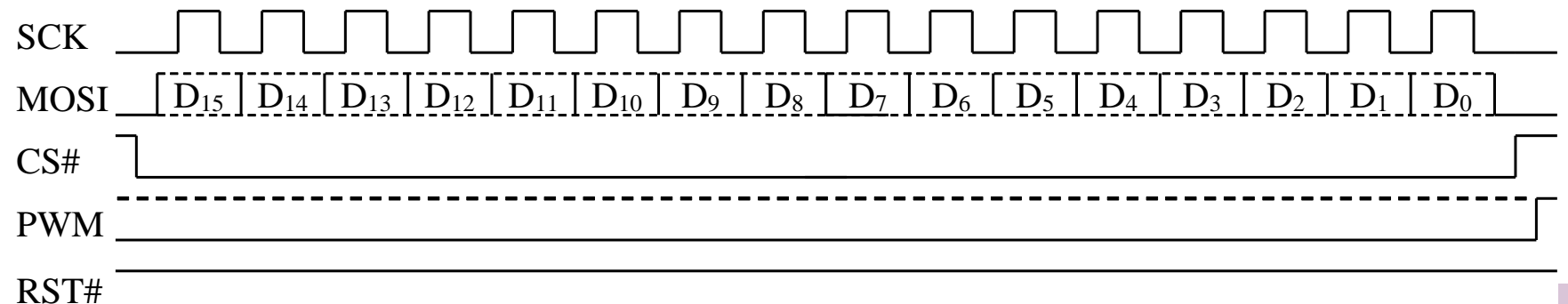
- Egyszerre egy adatbit
  - Sorban egymás után küldjük a biteket
- Az adatbit valamilyen órajelre változik
  - Számolni kell hányadik órajel jött eddig, az annyiadik bitet látjuk most
- Kell-e az órajelet is továbbítani?
  - Ha a másik oldalon „ugyanez” az órajel rendelkezésre áll, vagy elő tudjuk állítani akkor nem
    - ilyenkor „aszinkron soros adatátvitel”-ről beszélünk
  - Továbbíthatjuk is a jel mellett → „szinkron soros adatátvitel”

# Bitsoros adatátvitel sebessége

- Az adatátvitel sebességét az órajel frekvenciája határozza meg
  - Jelzés/másodperc → baud/sec, bps, „baud rate”
- Bináris jeltovábbítás esetén: baud/sec = bit/s
- Tipikus értékek aszinkron esetben: 300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 57600, 115200 bit/s
  - Szinkron esetben mindig nagyobb sebesség érhető el.
- Általában fizikai korátok (rendelkezésre álló sáv szélesség, zaj) korlátozzák a maximális átvihető sebességet
- Az átviteli sebesség beállítása programozható frekvencia osztóval történik
  - „Baud rate generátor”

# Szinkron soros adatátvitel

- Az átvitelt ütemező órajelet is átvisszük  
→ Plusz egy vezeték
- Az órajel meghatározott élénél érvényes az adat, a másik élénél pedig változhat.  
→ A kijelzőnél a laboron a felfutó élre volt érvényes és a lefutó környékén változott
- Jellemzően kell valamilyen jelzés, ami mutatja, ha egy új adat kezdődik.  
→ pl. a kijelző „protokolljában” ilyen a CS# jel



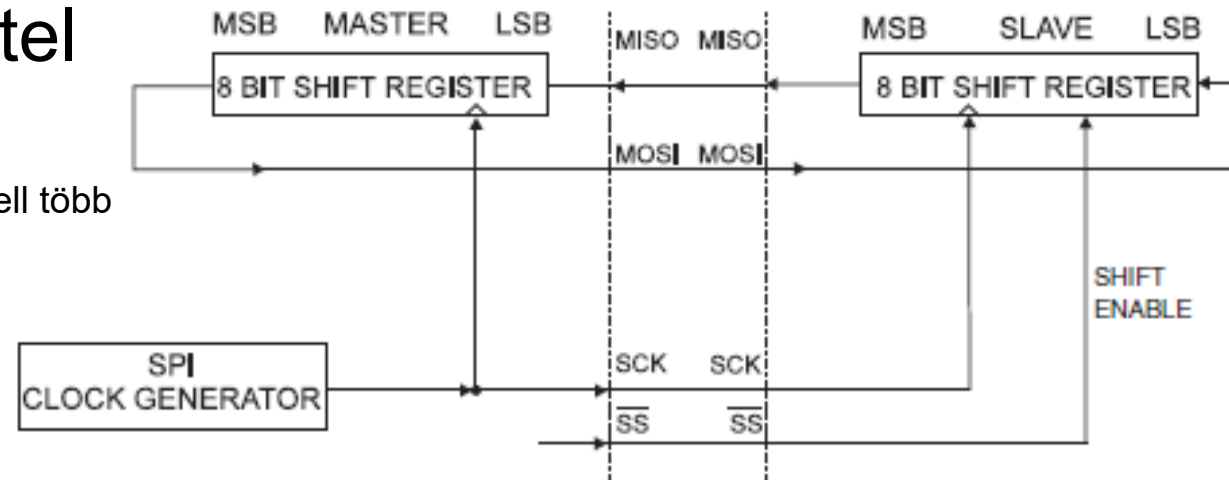
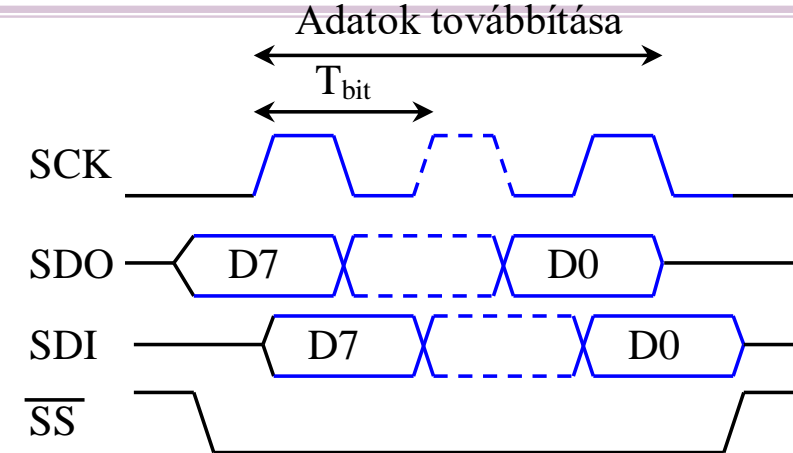
# Serial Peripheral Interface (SPI)

- Általános szinkron, full duplex, soros kommunikációs periféria
- Majdnem minden mikrokontrollerben van néhány ilyen
- Nagyon sok általános periféria eszköz (kijelzők, szenzorok, soros elérésű memóriák, stb.) használják.
- Mindenki használta már, mert igazából ezt írtuk meg a kijezõ kezeléséhez a 2. laboron.
- És egyébként is kb. mindenki minden nap használja, mert pl. az egyik legelterjedtebb adathordozó, az SD kártya is így kommunikál...



# SPI működése

- Az egyik eszköz irányítja az adatátvitelt  
→ Master
- A többi eszköz csak „válaszol”  
→ Slave
- A master a megfelelő SS (slave select) vezetéken választja ki az adott slave-et
- Az SS jel maga jelzi is az adatátvitel kezdetét/végét  
→ Több slave is lehet, az adat és órajel vezetékekből nem kell több
- Az órajelet is a master adja
- A konkrét átvitt bitek jelentése csak az adott eszköztől függ (a kijelző ugye kiírta...)



# SPI működése

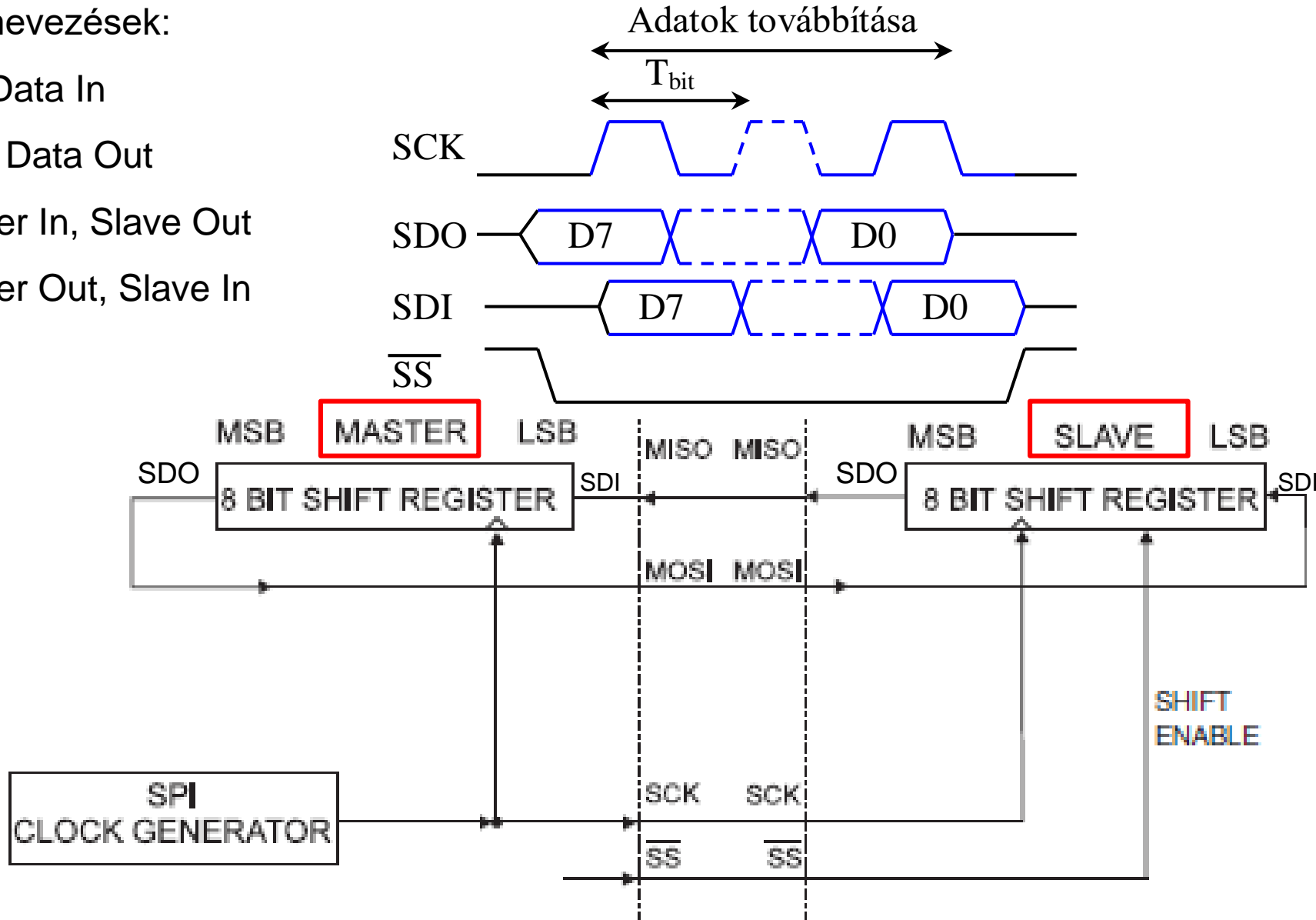
Jellemző elnevezések:

SDI: Serial Data In

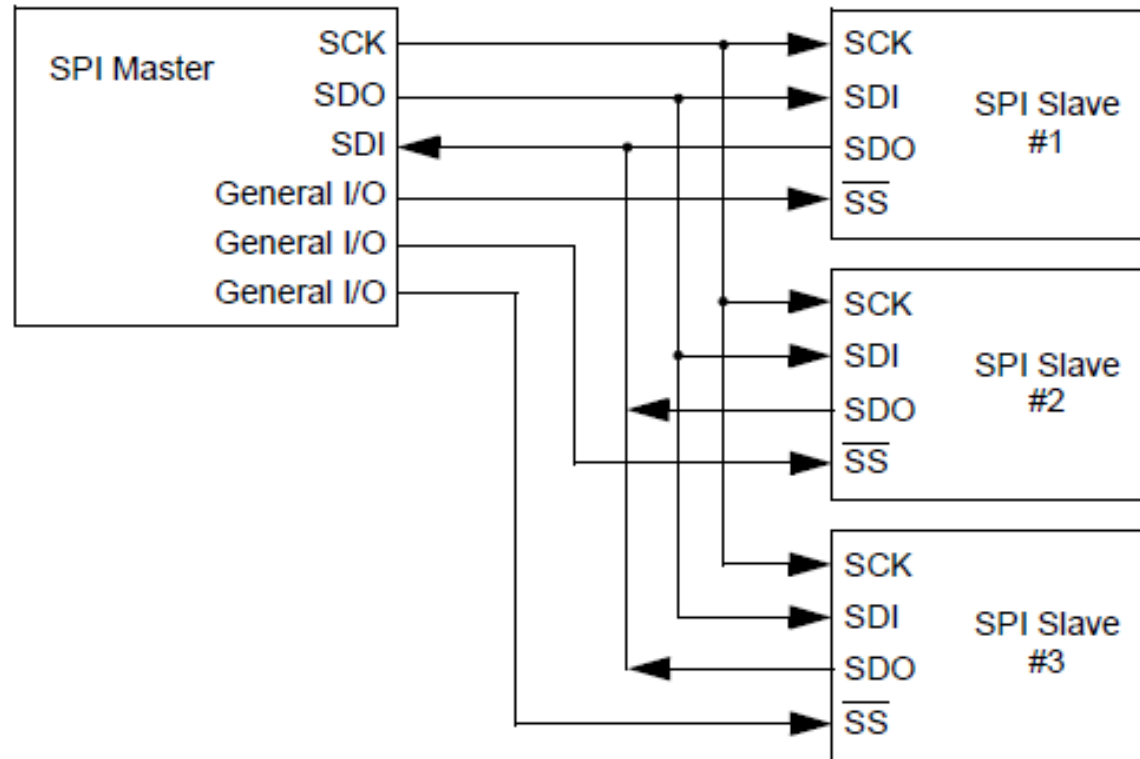
SDO: Serial Data Out

MISO: Master In, Slave Out

MOSI: Master Out, Slave In

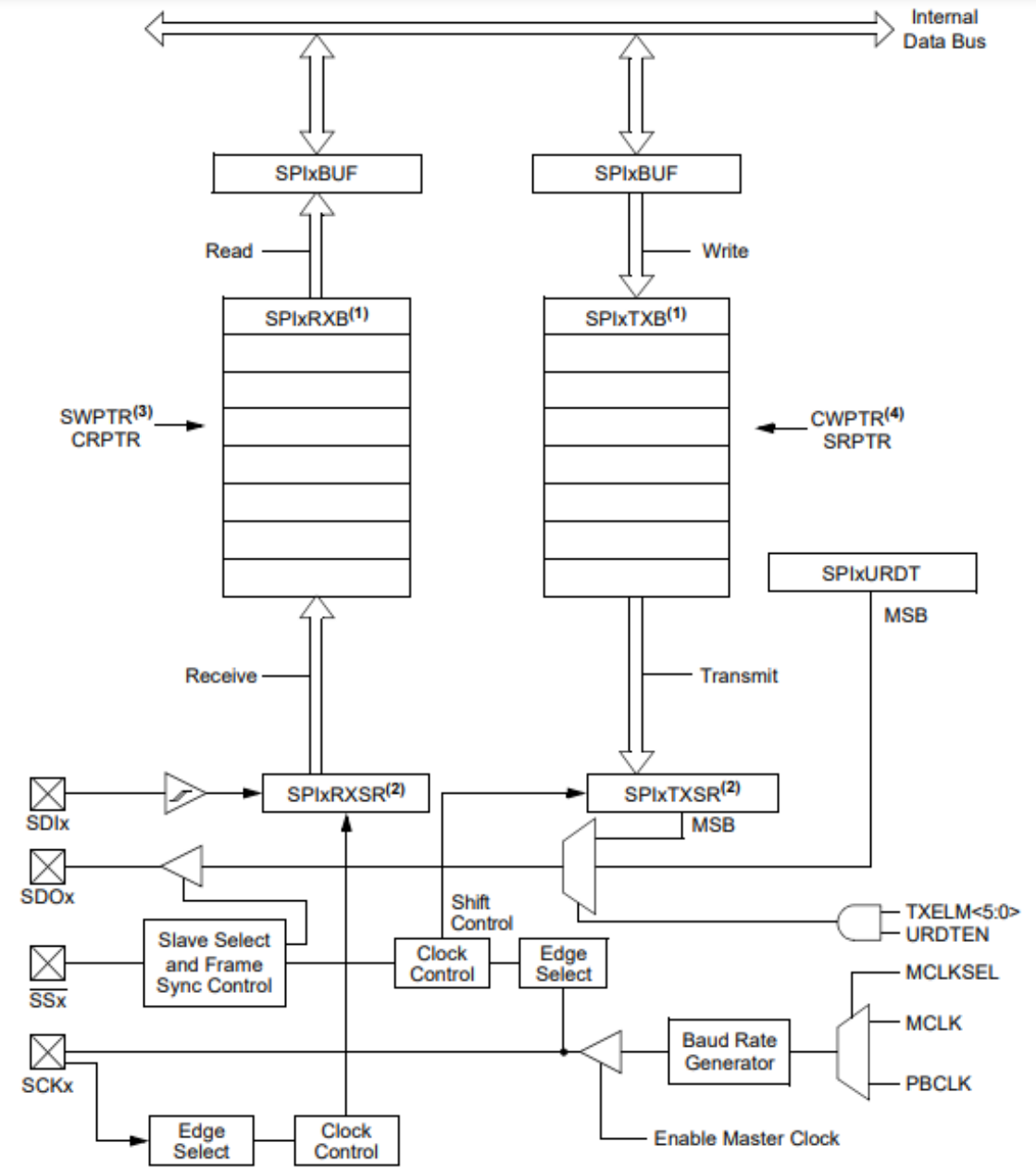


# Több SPI egység csatlakoztatása



# SPI hardver a PIC24-ben

- Külön adó és vevő shift-regiszterek
- Mindkettőhöz egy-egy FIFO kapcsolódik
  - ezt nem muszáj használni
- Választható órajel osztó („baud rate generator”)
- Az átvitel befejeződése megszakítást okozhat



- $$F_{SCK} = \frac{F_{PB}}{2 \cdot (SPIxBRG + 1)}, \text{ ahol } F_{PB}=16\text{MHZ}$$



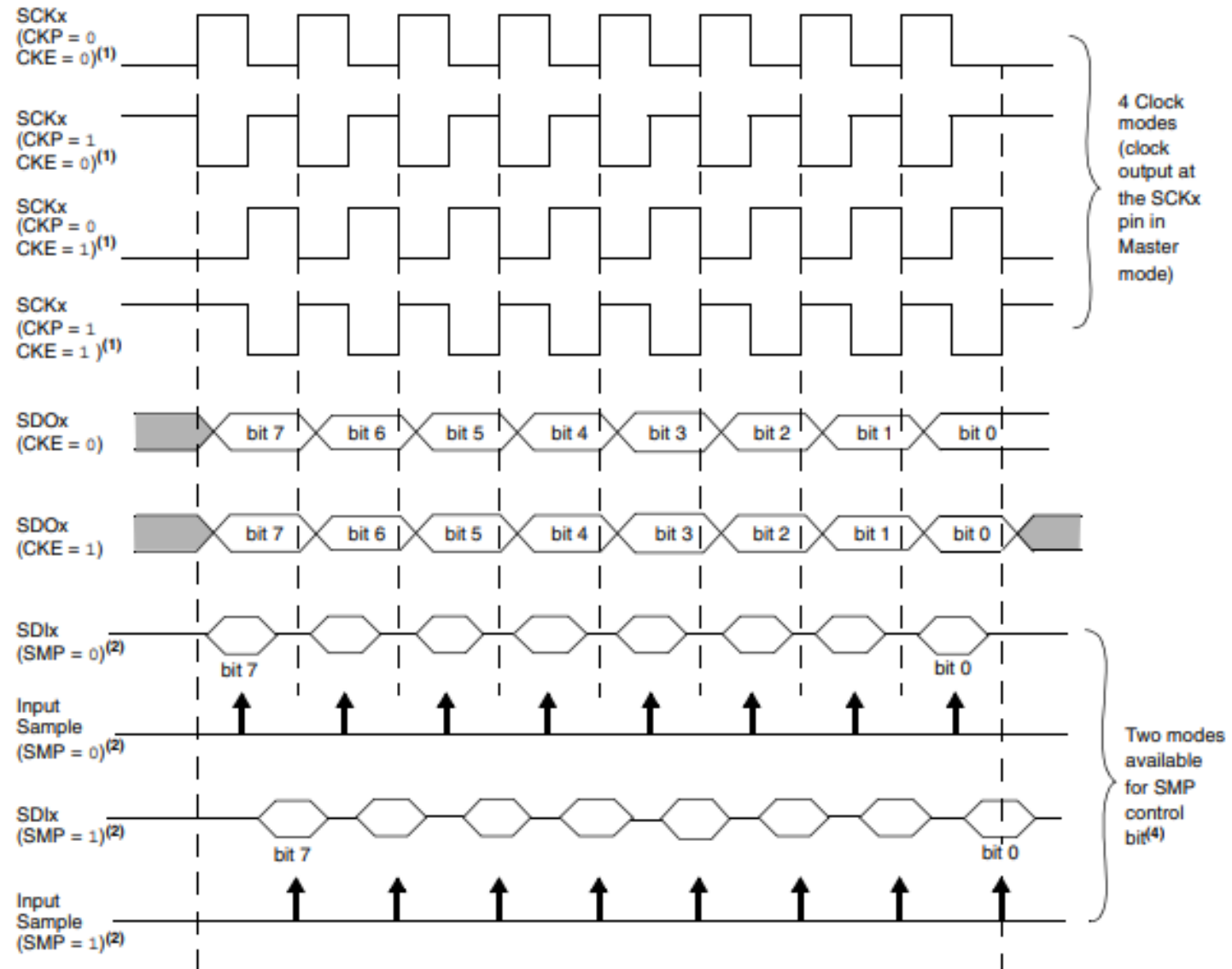
# Vezérlő regiszter

## SPI1CON1L:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
SPIEN	-	SPISIDL	0	0	MODE16	SMP	CKE	SSEN	CKP	MSTEN	0	0	MCLKEN	0	ENHBUF

<b>SPIEN</b>	SPI engedélyezés	1 = SPI engedélyezve 0 = SPI tiltva	<b>SSEN</b>	Slave select láb használata (csak slave módban!)	1 = Slave módban az SS bemenet 0 értéke esetén működik csak az SPI 0 = Az SS bemenet nincs használva, az SPI folyamatosan működik
<b>SPISIDL</b>	Működés IDLE módban	1 = Az SPI IDLE módban leáll 0 = Az SPI IDLE módban is működik	<b>CKP</b>	Órajel polaritás választás	1 = Az órajel aktív állapota alacsony, inaktív állapota magas. 0 = Az órajel aktív állapota a magas, inaktív állapota az alacsony szint.
<b>MODE16</b>	16/8 bites adat mód	1 = a küldött adat 16 bites 0 = a küldött adat 8 bites	<b>MSTEN</b>	Üzem mód	1 = Master 0 = Slave
<b>SMP</b>	Bemenet mintavételezés ideje	1 = A bemenet mintavételezése a kimenet változtatásakor történik 0 = A bemenet mintavételezése két kimenet változás között közepén történik.	<b>MCLKEN</b>	Órajel forrás választás	1 = Az SPI órajele a külső busz órajel kimenet (REFO, utasításciklus ideje) 0 = Az SPI órajele a belső órajelből lesz leosztva (a BRG-vel)
<b>CKE</b>	Órajel él választás	1 = A kimenet az órajel inaktívból aktívba váltásakor stabil 0 = A kimenet az órajel aktívból inaktívba váltásakor stabil	<b>ENHBUF</b>	Bővített buffer mód	1 = Az adó és vevő bufferek (32 bájtos) FIFO sorok 0 = Az adó és vevő buffer egyetlen regiszter (FIFO kikapcsolva)

# SPI üzemmódok



# Állapot regiszter

SPI1STATL:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
U-0	U-0	U-0	R/C-0	R-0	U-0	U-0	R-0	R-0	R/C-0	R-1	U-0	R-1	U-0	R-0	R-0
-	-	-	FRMERR	SPIBUSY	-	-	SPITUR	SRMT	SPIROV	SPIRBE	-	SPITBE	-	SPITBF	SPIRBF

<b>FRMERR</b>	Kerethiba	SPI keret hiba történt csak slave módban, SS előbb ment fel, minthogy a megfelelő számú adat érkezett	<b>SPIRBE</b>	Vételi buffer üres	1 = A vételi buffer üres 0 = A vételi buffer nem üres (olvasható)
<b>SPIBUSY</b>	SPI foglalt	1 = Adatátvitel folyamatban 0 = A periféria épp áll	<b>SPITBE</b>	Adó buffer üres	1 = Az adó buffer üres (írható) 0 = Az adó bufferben van valami
<b>SPITUR</b>	Adatelfogyás hiba	1 = Nem volt mit küldeni, amikor olvastak (csak slave módban) 0 = nem történt ilyen hiba	<b>SPITBF</b>	Adó buffer tele	1 = Az adó buffer tele 0 = Az adó buffer nincs tele
<b>SRMT</b>	Adás vége jelzés	1 = Az utolsó adat is kiküldésre került a bufferből 0 = Adatküldés van folyamatban	<b>SPIRBF</b>	Vételi buffer tele	1 = A vételi buffer tele van 0 = A vételi buffer nincs tele
<b>SPIROV</b>	Túlfutás	1 = Adatvesztés történt, mert adat érkezett, de a vételi bufferben nem volt hely 0 = nem volt ilyen hiba			



- Az órajelet nem visszük át külön vezetéken
  - Mindkét oldalon rendelkezésre kell álljon „ugyan az” az órajel
    - Egyforma frekvencia
    - Egyforma fázishelyzet

*Ha a két órajel egyforma frekvenciájú, de „nem egyszerre kezdődött”, nem azonos pillanatban jönnek az élváltások, vagyis nem azonos a fázishelyzetük.*
  - A „vevő” oldalnak az órajelet szinkronizálni kell az adó oldal órajelével.
    - Vagyis tudni kell, hogy az adó oldal órajele „mikor kezdődött”
    - Emiatt további jelzések szükségesek
    - Ez a szinkronizáció általában megoldja az adat kezdetének jelzését is

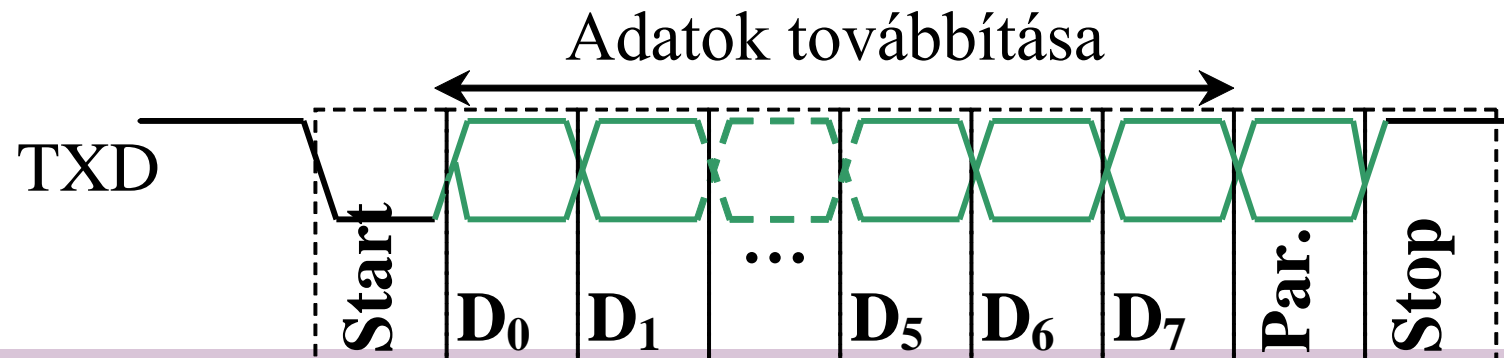
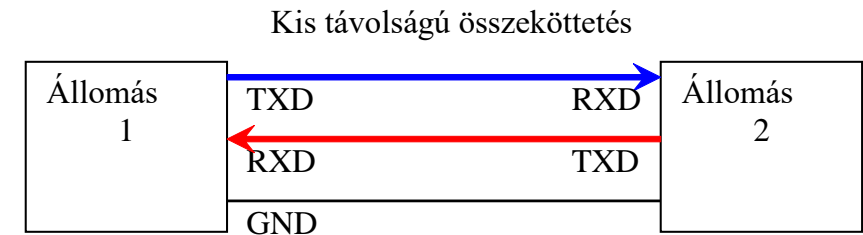
- Általános célú, aszinkron, full duplex kommunikációs periféria

*könnyű belőle half duplex-et vagy simplex-et alakítani, ha kellene*

- Majdnem minden mikrokontrollerben van néhány ilyen
- Nagyon sok általános periféria eszköz (kijelzők, szenzorok, stb) használják.
- A konkrét logikai jeleket több különböző szabványos ú.n. fizikai rétegen lehet átvinni (pl. EIA/RS232, EIA422, EIA485, HART, stb.)
- Sok elterjedt kommunikációs szabvány alapja  
(Profibus, Modbus, telefonmodemek, GPS vevők, unix/linux terminál, stb.)

# UART működése

- Bitsoros adattovábbítás egyetlen jelvezetéken
- Full duplex:
  - Adó kimenet: TXD, vevő bemenet: RXD
- Vonal nyugalmi állapota: „1”
  - A lefutó él kezdi az adatátvitelt
    - Itt szinkronizálódnak a két eszközben „szabadon” járó órajelek
- Adatbitek keretezése: Start(0), adatbitek, STOP(1)
  - LSB (least significant bit) először
  - általában 8 bitet küldünk egyszerre, de néha van ettől kevesebbre és többre is lehetőség
- Opcionális paritás bit az adatbitek után (Páros vagy Páratlan) → hibadetektálás
  - A paritásbit az adatban lévő 1-esek számát egészíti ki párosra vagy páratlanra beállítástól függően

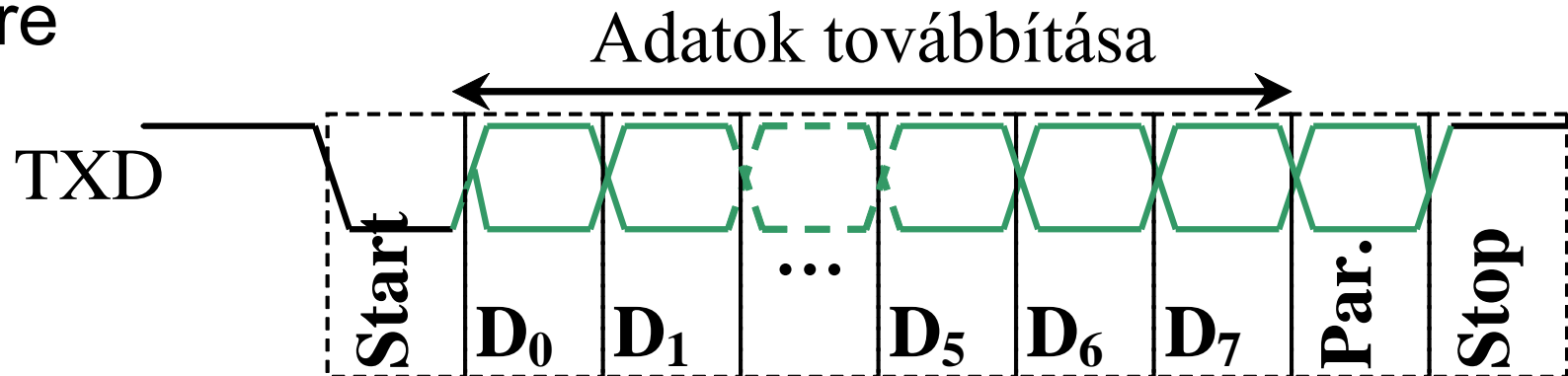


Vevő oldalon észrevehető hibák:

- paritás: nem megfelelő érték van a paritásbit helyén
- keret(frame): A stop bit helyett még mindig 0 volt
- túlfurás(overrun): új adat érkezett, mielőtt az előzőt feldolgoztuk

- BREAK jelzés

- Annyi hosszú 0-t küld, hogy az biztosan kerethibát okozzon
- Szándékos hibajelzésre



- Az adatátvitelhez feltétlenül szükséges RX és TX jelek mellett sok UART alkalmazás további jeleket is használ

*Hagyományosan az analóg telefonmodemek céljára találták ki ezeket*

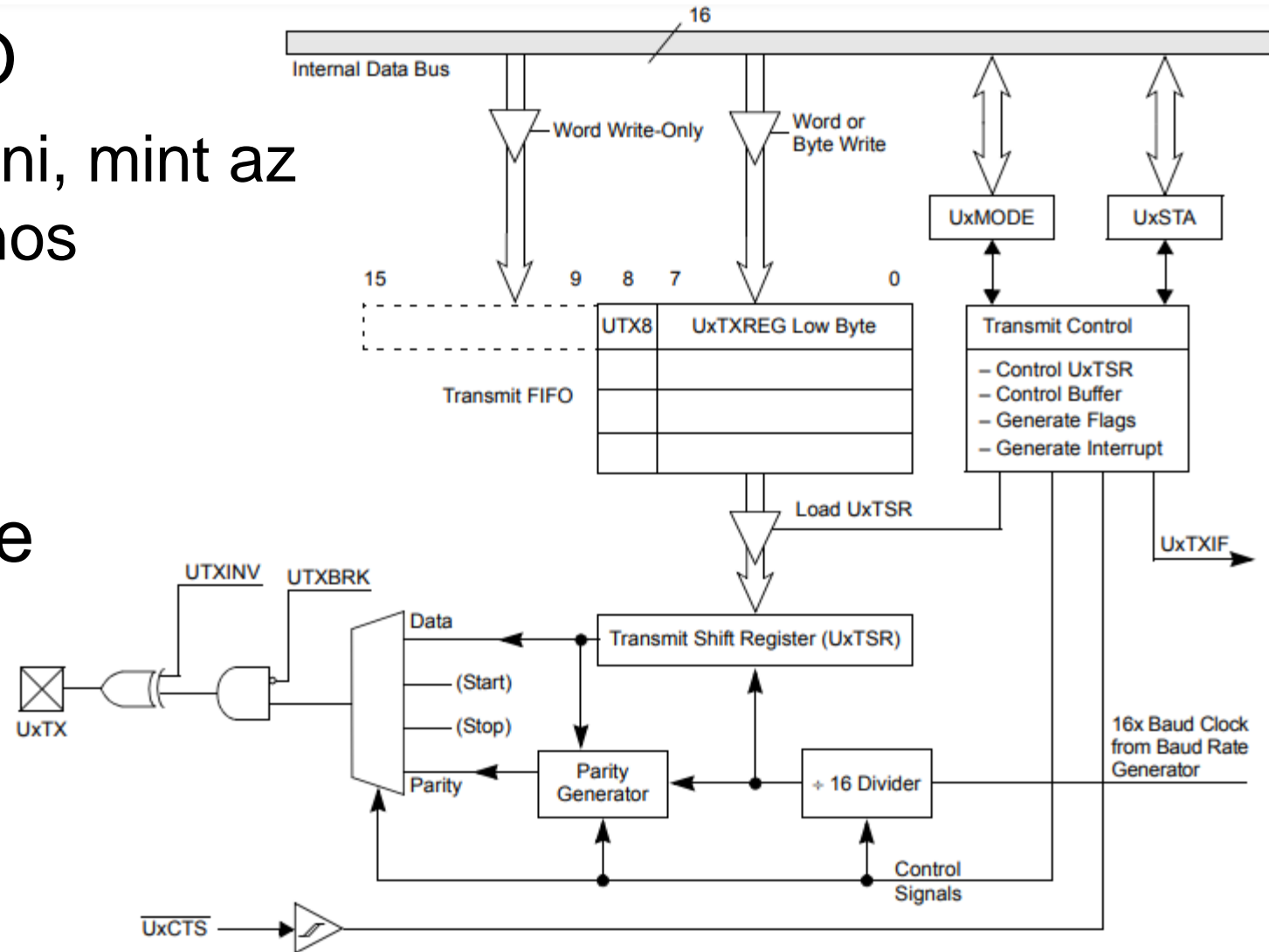
- 2 jelpár:

- DTR: data terminal ready
  - Eredetileg ezzel jelezte a számítógép a modemnek, hogy készen áll (be van kapcsolva)
- DSR: data set ready
  - Eredetileg ezzel jelezte a modem a számítógépnek, hogy be van kapcsolva
- RTS: request to send
  - Az adó oldal jelzése, hogy adni szeretne valamit (szó szerint: „szeretnék adni”)
- CTS: clear to send
  - A vevő oldal jelzése, hogy készen áll egy adat vételére (szó szerint: „szabad adnod”)

- Minden UART modul tartalmaz:
  - Baud rate generátort
    - órajelosztó, itt az adó és vevő oldalon is kell
  - Adó hardver egységet
  - Vevő hardver egységet
- „Modem vezérlő” jelek-et kezelő hálózatot

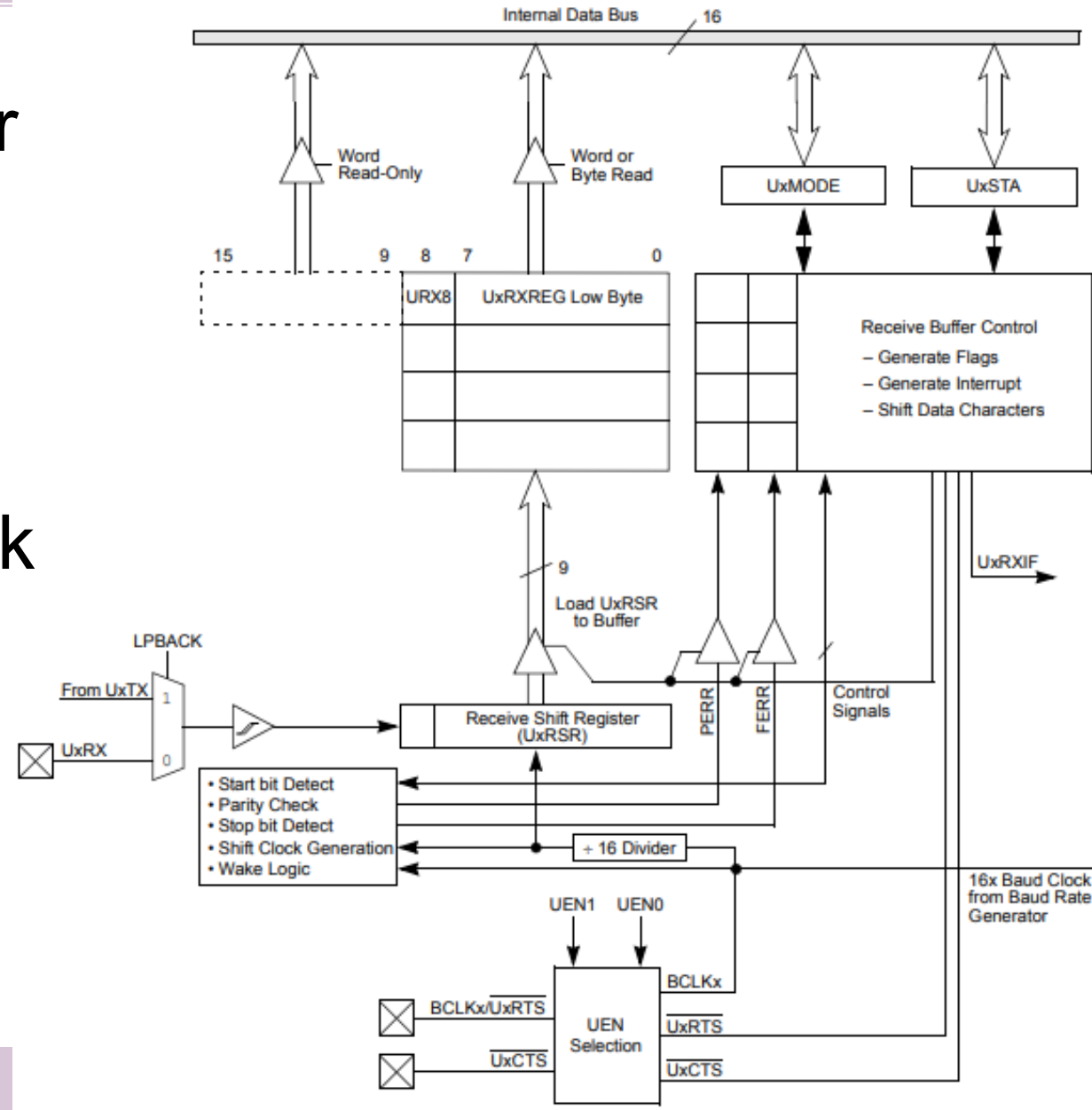
# UART adó a PIC24-ben

- Fixen 4 bájtos adó FIFO
  - ezt nem lehet kikapcsolni, mint az SPI-ben, de nagyon hasznos
- Lehetőség a kimenet invertálására
- CTS figyelembe vételére



# UART vevő a PIC24-ben

- Fixen 4 bájtos vevő FIFO puffer
- Lehetőség a saját adó kimenet „visszahallgatására”
- Hiba ellenőrzés
- Megszakítás kérési lehetőségek





- UxRXREG
  - Vett adatok helye (csak olvasható)
  - Igazából a FIFO „teteje”  
(vagyis a legrégebben beérkezett, még ki nem olvasott bájtt, *ami még belefér a FIFO-ba*)
- UxTXREG
  - Kimenő adatok helye (csak írható)
  - Igazából a FIFO „alja”  
(vagyis ide lehet új adatot írni, de lehet, hogy vannak benne már adatok)
- UxMODE: Beállításokat tartalmazó regiszter
- UxSTA: Állapotjelző biteket tartalmazó regiszter (*és néhány beállítás is csak ide fért el*)
- UxBRG: Baud rate generátor osztó regisztere

# UxMODE regiszter

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W-0	U-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
UARTEN	-	USIDL	IREN	RTSMOD	-	UEN1	UEN0	WAKE	LPBACK	ABAUD	URXINV	BRGH	PDSEL1	PDSEL0	STSEL
UARTEN	indítás/leállítási		1 = UART engedélyezve 0 = UART tiltva				WAKE	„Felébresztő megszakítás”		1 = Az UART RX bemenetén érkező lefutó él azonnal megszakítást okoz (a processzor azonnal felébreszthető kommunikáció kezdetekor) 0 = Csak tényleges adat vétele okoz megszakítást					
USILD	Működés IDLE módban		1 = Az UART IDLE módban leáll 0 = Az UART IDLE módban is működik				LPBACK	Teszt mód		1 = A kiküldött adatok visszaérkeznek, mintha vett adatok lennének 0 = Normális működés					
IREN	IRDA vevő mód		1 = IRDA (infravörös távirányító protokoll) mód 0 = „hagyományos” UART mód				ABAUD	Baud rate automatikus megállapítása		1 = A következő karakter vételéből megméri a baud rate-et (csak 0x55 karakterrel működik). A mérés után a bit 0-ba vált.					
RTSMOD	RTS kimenet módja		1 = Adás engedélyezés mód 0 = Hagyományos RTS mód				URXINV	Vétel láb polaritás megfordítása		1 = RX nyugalomban '0' 0 = RX nyugalomban '1' (hagyományos UART)					
UEN [1:0]	UART-hoz tartozó portlábak engedélyezése		11: TX, RX és órajel 10: TX, RX, RTS és CTS 01: TX, RX és RTS 00: csak TX és RX				PDSEL	Adathossz és paritás állítás		11 = 9 bites adat, nincs paritás 10 = 8 bites adat, páratlan paritás 01 = 8 bites adat, páros paritás 00 = 8 bites adat, nincs paritás					
BRGH=0							BRGH=1		STSEL	Stop bit hossza		1 = 2 db stop bit 0 = 1 db stop bit			

$$\text{Baud Rate} = \frac{F_{CY}}{16 \cdot (UxBRG + 1)} \quad \text{BRGH=0}$$

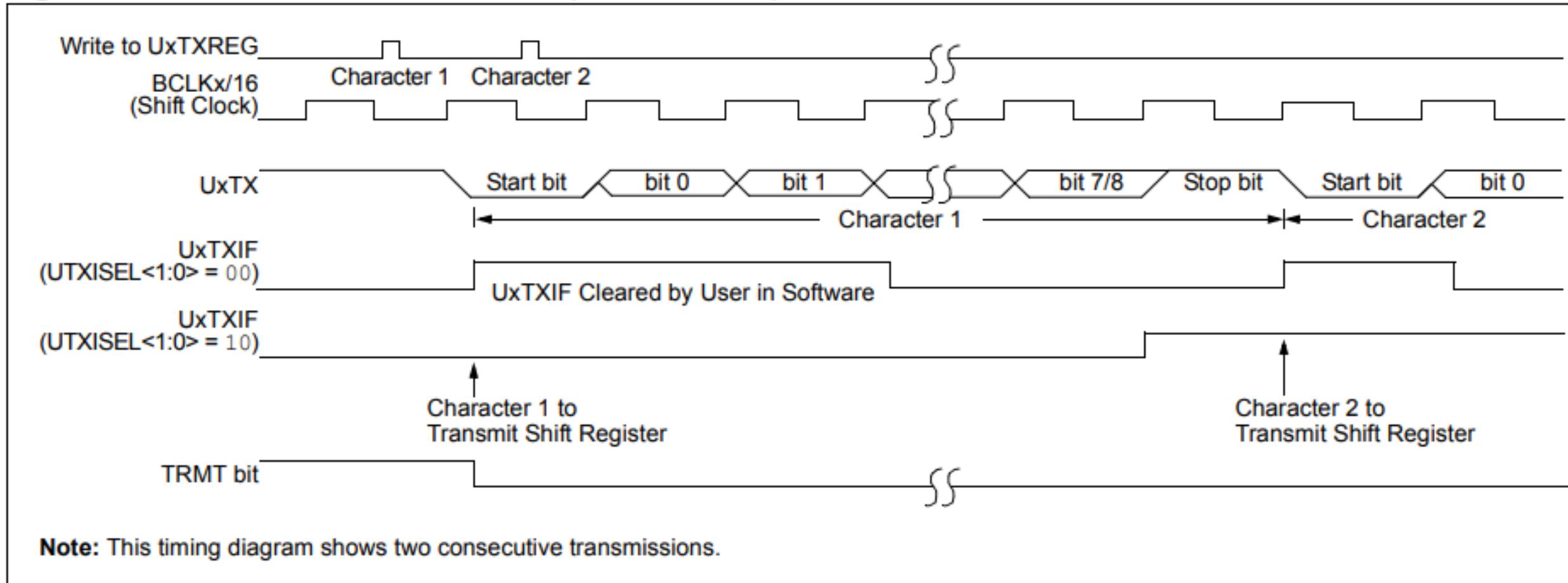
$$\text{Baud Rate} = \frac{F_{CY}}{4 \cdot (UxBRG + 1)} \quad \text{BRGH=1}$$



15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R-1	R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0
UTXISEL1	UTXINV	UTXISEL0	URXEN	UTXBRK	UTXEN	UTXBF	TRMT	URXISEL1,0		ADDEN	RIDLE	PERR	FERR	OERR	URXDA
<b>UTXISEL [0:1]</b>	Adási megszakítás kérésének módja	11 = Érvénytelen Megszakítást kér, ha: 10 = Egy beírt adat küldése elkezdődött 01 = Minden küldés befejeződött 00 = Van hely a küldő bufferben (beírhatjuk a következő adatot)					<b>URXISEL [0:1]</b>	Vételi megszakítás kérésének módja		Megszakítást kér, ha: 11 = A vételi buffer tele van (további vétel adatvesztést okozna) 10 = A vételi bufferben már csak 1 adatnak van hely 0- = Van ki nem olvasott adat a vételi bufferben					
<b>UTXINV</b>	Adás láb polaritás megfordítása	1 = TX nyugalomban '0' 0 = TX nyugalomban '1' (hagyományos UART)					<b>ADDEN</b>	9. bit címként értelmezése		1 = 9 bites módban a 9. bit 1-es értéke esetén az adatot címként értelmezi 0 = nem kezeli speciálisan a 9. bitet					
<b>URXEN</b>	Vétel engedélyezés	1 = Vétel engedélyezve 0 = Vétel tiltva					<b>RIDLE</b>	Nincs vétel folyamatban		1 = Nincs vétel folyamatban 0 = Éppen vétel van folyamatban					
<b>UTXBRK</b>	BREAK küldés	1 = BREAK karakter küldése, automatikusan 0-ba áll ha befejeződött					<b>PERR</b>	Paritáshiba		1 = A vételi bufferben aktuálisan olvasható adat paritáshibás volt 0 = nem volt ilyen hiba					
<b>UTXEN</b>	Adás engedélyezés	1 = Adás engedélyezve 0 = Adás tiltva					<b>FERR</b>	Kerethiba		1 = A vételi bufferben aktuálisan olvasható adat keretezés hibás volt 0 = nem volt ilyen hiba					
<b>UTXBF</b>	Adó buffer tele jelzés	1 = Az adó buffer tele 0 = Van hely az adó bufferben					<b>OERR</b>	Túlfutás		1 = Adatvesztés történt, mert adat érkezett, de a vételi bufferben nem volt hely 0 = nem volt ilyen hiba					
<b>TRMT</b>	Adás vége jelzés	1 = Nincs adás folyamatban, minden korábbi küldés befejeződött 0 = Adás folyamatban					<b>URXDA</b>	Vett adat elérhető		1 = A vételi pufferben van olvasható adat 0 = A vételi puffer üres					

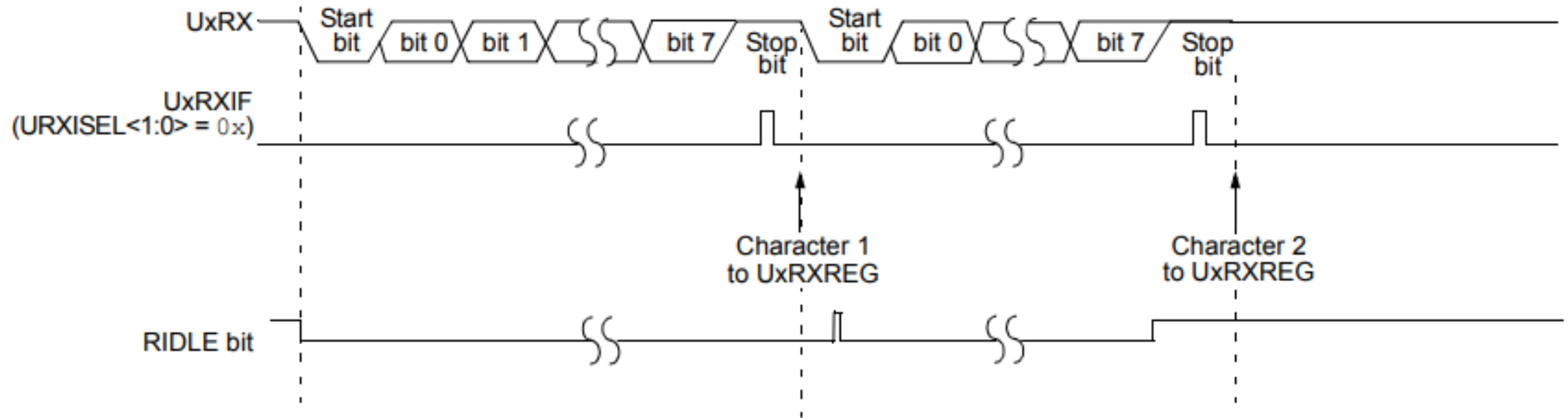
# Adás idődiagramja

**Figure 5-4: UARTx Transmission (Back-to-Back)**



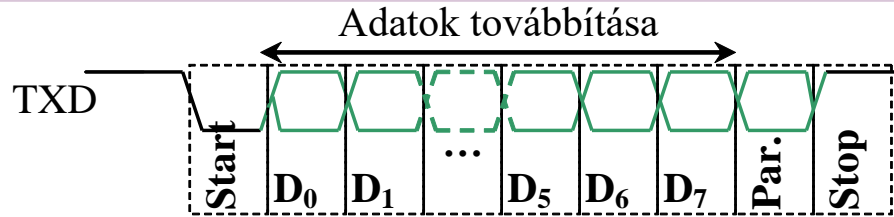
# Vétel idődiagramja

**Figure 7-2: UARTx Reception<sup>(1,2)</sup>**

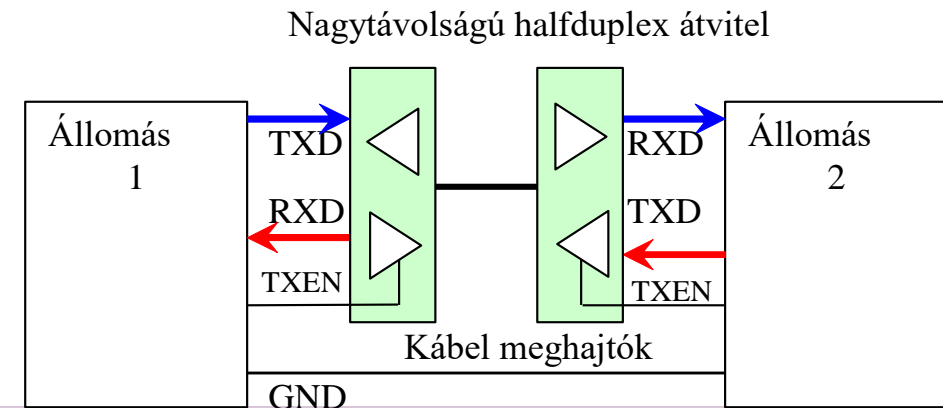
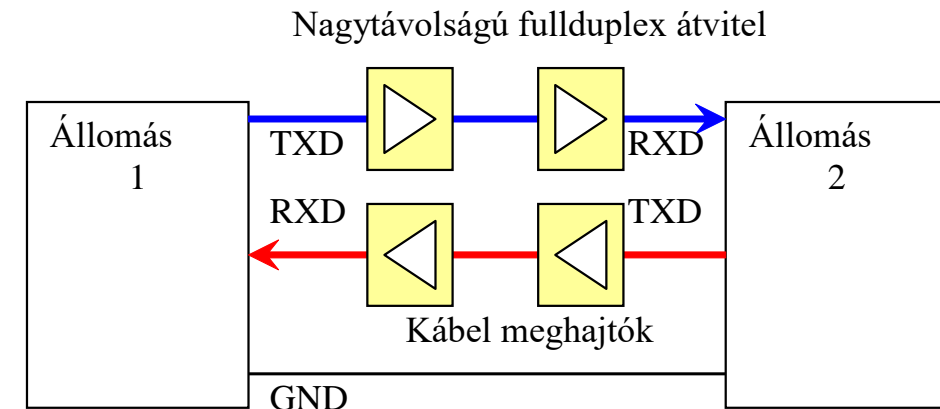
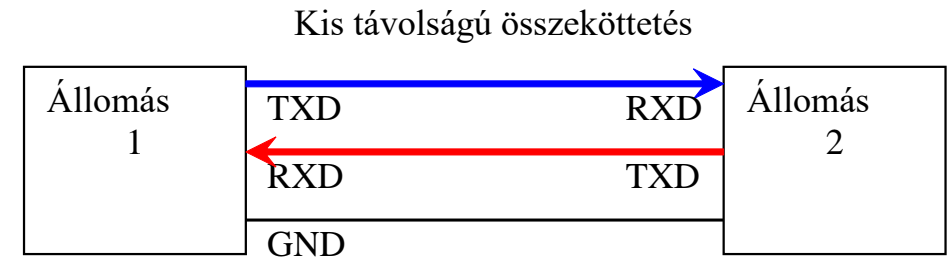


- Note 1:** This timing diagram shows two characters received on the UxRX input.
- Note 2:** If the interrupt flag is used by the application software as a basis for disabling the UART transmission, the software should wait for 1-bit time before disabling the transmission.

# UART fizikai rétegek



- A portlábakon kiadott logikai jelet nem lehet „akármilyen messzire” elvinni
- Kis távolságra közvetlen összeköttetés
- Nagyobb távolságra meghajtó áramkör kell
- A meghajtó lehet a kiépítéstől függően:
  - Simplex,
  - halfduplex,
  - Fullduplex
- Nagy távolságra szokás még halfduplex, szimmetrikus (differenciális) jelátvitel



# UART fizikai rész

- Akár több adó/vevő busz jellegű összeköttetése is lehetséges
- Többféle elterjedt szabvány, pl.:
  - TIA/EIA 232: full duplex, 2 vezetéken "PC soros port, RS232"
  - TIA/EIA 485: half duplex, 2 vezetéken, differenciálisan
  - TIA/EIA 422: full duplex, 4 vezetéken, differenciálisan

TIA: Telecommunications Industry Association

EIA: Electronic Industries Alliance

Interfész	Adók száma	Vevők száma	„0” fizikai jelszintje	„1” fizikai jelszintje	Max. távolság
TIA/EIA232	1	1	+3..+15V	-3..-15V	15m
TIA/EIA422	1	10	>+200mV	<-200mV	1200m
TIA/EIA485	32	32	>+200mV	<-200mV	1200m

- A legtöbb kommunikációs interfész a fenti kettő (UART vagy SPI alapján) megérthető.
- Jellemző hasonlóság:
  - Bitsoros továbbítás
  - Shift regiszter(ek)+vezérlő hálózat
- Jellemző különbségek:
  - Címzés lehetősége (pl. egy kommunikációs csatornán több eszköz).
  - Az adatátvitel elejének és végének jelzésére szolgáló egyéb megoldások.
  - Az adatátvitel hibaellenőrzésére szolgáló plusz megoldások.



(A teljesség igénye nélkül)

- I2C

- Kis távolságra (berendezésen belül használatos)
- 2 vezeték (adat, órajel) → szinkron
  - Az adat az órajel 1 értéke alatt nem változik
- Az adatátvitel kezdetét/végét speciális jelzéssel oldja meg:
  - Az órajel 1-es értéke alatti lefutó él az adaton a START feltétel
  - Az órajel 1-es értéke alatti felfutó él az adaton a STOP feltétel
- Az adat vezeték kétirányú (és az órajel is lehet az)
- A két irányú meghajtást open collector jelekkel oldja meg.
- Címzést is lehetővé tesz (az első elküldött 7 vagy 10 bit a cím)
  - Egy master egység több slave egységgel is kommunikálhat. (persze csak felváltva)
  - Több master egység is lehet.

(A teljesség igénye nélkül)

- Dual/Quad SPI / SDIO
  - Lényegében SPI, de több (2 vagy 4) párhuzamos adatvonallal
  - Szinkron
  - Egy master és egy slave összekötésére alkalmas
    - Esetleg lehet „CS” jel, de itt nem szokták használni
  - Az adatvonalak kétirányúak → csak halfduplex lehet.
  - Kis távolságra, de nagyon nagy sebesség  
(~200MHz-es órajel, 4 bit párhuzamosan → Gbit/s nagyságrendű sebesség)

(A teljesség igénye nélkül)

- CAN

- Nagyobb távolságra (<40m)
- Először járműipari célra (autó fedélzeti rendszereinek összekötésére)
- Több master (minden csomópont küldhet és fogadhat)
- Csak két „összetartozó” adatvonal (CANH, CANL)
  - Egyszerű logika:
    - Az adatvonalak között ellenállások „összehúzzák” a két vonalat. → recesszív szint (logikai 1)
    - Az adó, ha adni akar:
      - 1 esetén nem csinál semmit
      - 0 esetén „szét” húzza a két vonalat (egyiket táp fele, másikat föld fele) → domináns szint (logikai 0)
  - Half duplex (egyszerre csak egy eszköz adhat)
    - Meg van oldva, hogyha két csomópont egyszerre küldene valamit akkor sem veszik adat
- „Aszinkron”, nincs továbbított órajel, a továbbított adatokból tud szinkronizálni

(A teljesség igénye nélkül)

- USB

- Elsősorban PC perifériákhoz
- Nagyon sok változata van
  - A különböző sebességű változatok nem „csak” sebességben különböznek de egymással mindig visszafele kompatibilisek
- Nincs külön továbbított órajel
- Differenciális jel továbbítás
  - D+,D-:
    - Adat és negáltja a normál adatok esetében
      - „1-es”: egyik alacsony, másik magas, „0-s”: egyik magas, másik alacsony
  - Az adatátvitel elejét, végét és a hibákat a „szabály” megszegése jelenti:
    - Mindkét adatvonal alacsony: Adatátvitel vége
    - Mindkét adatvonal magas: Hiba

