

## Mikro počítačové Systémy

### MIPS

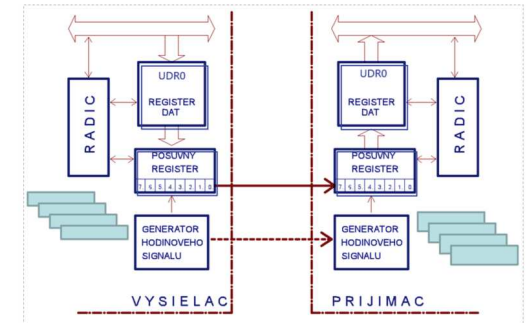
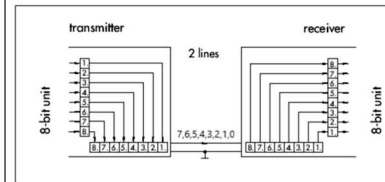
Distribuované vnorené počítačové systémy  
Distributed Embedded Computer System  
(Microcontrollers)

## Prednáška 7. Sériový prenos údajov.

*Ktorý prenos je rýchlejší: Paralelný alebo sériový?  
Prifázovanie.*

1

## Sériový prenos údajov.



Výhody: Menší počet vodičov  
Nevýhody: „rozloženie“- „zloženie“  
Obvod: USART (Universal Synchronous Asynchronous Receiver Transmitter)

2

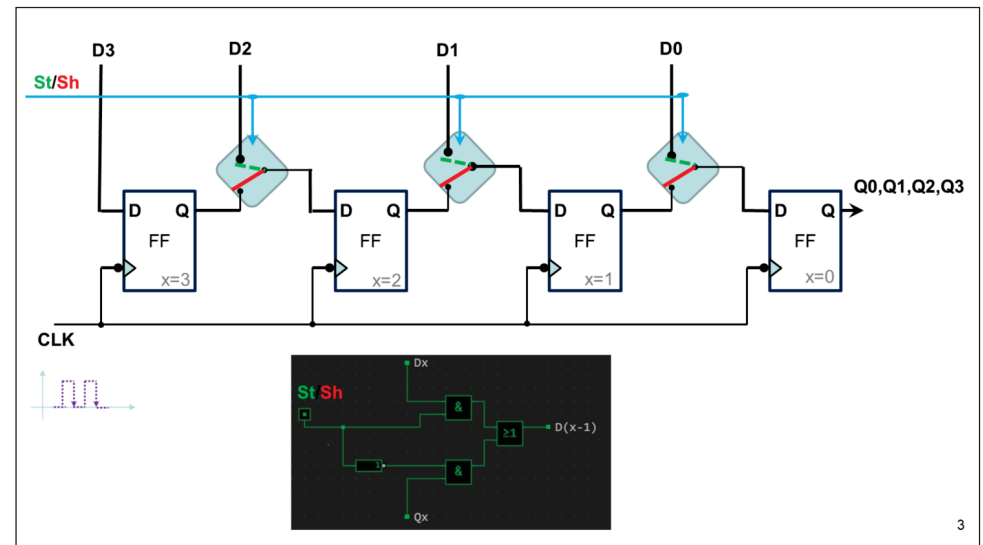
Dalo by sa povedať, že paralelný. Veď súčasne prenášame viacero informácií. Potom je namiesto otázka prečo sa v PC používa sériová zbernica PCI Express?

Prifázovanie je proces pripojenia alternátora do elektrickej siete. Pritom musí byť splnený celý rad podmienok pre zdroj a sieť: Rovnaké napätie, frekvencia, rovnaké poradie fáz, rovnaká fáza.

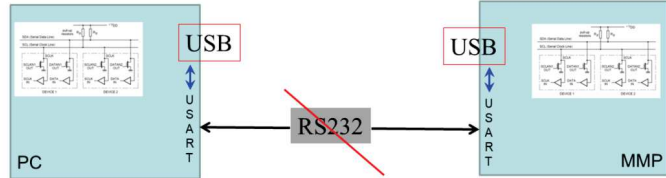
Podobné úlohy treba riešiť aj pri sériovom asynchrónom prenose.

Sériová rozhranie a sériová komunikácia patrí k najrozšírenejším metódam komunikácie medzi počítačmi a počítačom a napr. MMP. Dáta sú pri sériovej komunikácii prenášané bit po bite za sebou, na rozdiel oproti paralelnému spôsobu komunikácie, kde sú informácie prenášané v určitých skupinách, napríklad bajtoch, naraz. Hlavná výhoda sériového prenosu dát oproti paralelnému je v tom, že na prenos informácie je potrebný menší počet vodičov. Nevýhody vyplývajú z toho že počítače používajú bajtový spôsob adresácie dát. Všetky dáta sú v pamäti uložené po bajtoch. Preto je potrebný obvod, ktorý rozloží dáta z paralelnej podoby na sériovú, vyšle ich bit po bite do komunikačného kanála a na druhej strane ten istý obvod urobí transformáciu zo sériovej podoby späť na paralelnú. Takýmto obvodom je USART (Universal Synchronous Asynchronous Receiver Transmitter). Voľakedy sa vyrábal

pod označením 8251. Ďalšia z jeho funkcií je, že signál, ktorý doňho vstupuje dopĺňa o signály ktoré kontrolujú proces vysielania a príjmu.



## Sériový prenos údajov.



### 1. Znaký a znakové reťazce (stringy)

Pod znakom rozumieme: čísllice, písmená a tzv. špeciálne znaky, napr. ?, !, \$, CR, LF atď. **Ak má znakom rozumieť aj počítač ??**, treba ich zakódovať. Kódovanie:

Priradenie binárneho čísla (byte, word) bližšie neurčeným spôsobom k znaku.

Pôvodné kódy boli krátke:

- Morseho kód.
- BCD - 4 b kód.
- Baudotov - 5 b kód. (Franc. ing.- pracoval s telegrafmi.)
- Fieldata - 6 b kód.
- ASCII - 7 b kód (celkovo 128 znakov) (American Standard Code for Information Interchange)
- EBCDIC 8b (Extended Binary Coded Decimal Interchange Code (IBM))
- UNICODE 16b

Binary	Letter	Figure
0000	A	0
0001	B	1
0010	C	2
0011	D	3
0100	E	4
0101	F	5
0110	G	6
0111	H	7
1000	I	8
1001	J	9
1010	K	10
1011	L	11
1100	M	12
1101	N	13
1110	O	14
1111	P	15
0000	Q	16
0001	R	17
0010	S	18
0011	T	19
0100	U	20
0101	V	21
0110	W	22
0111	X	23
1000	Y	24
1001	Z	25
1010	[	26
1011	\	27
1100	]	28
1101	^	29
1110	_	30
1111	~	31

USART vysiela a prijíma dátové rámce (data frames) ktoré sú tvorené z:

- 1 Start bit
- 5, 6, 7, 8 alebo 9 dátových bitov.
- Žiadnou, párnou, nepárnou... paritou. (no, even, or odd parity bit)
- 1 alebo dva stop bity.

4

USART by sme si nemali zamieňať s RS232 rozhraním (interface). USART zabezpečuje komunikačný protokol, a RS232 určuje úroveň signálov, dátových a riadiacich. V PC je RS232 už minulosť, ale prax toto rozhranie ešte stále akceptuje.

AVR má úrovne signálov 5V, resp. 3,3V pre log. 1/0 a 0V pre log. 0/1.

RS232 má podľa normy +3V až 25V pre log. 0/1 a -3 až -25V pre log. 1/0. **Riadiace signály sú v priamej logike a dáta v inverznej.** Posun úrovni zabezpečuje obvod, napr. MAX 232. Na doske

ARDUINO UNO máme obvod FT232 ktorý realizuje SPOJENIE USB a TTL.

Po nainštalovaní odpovedajúcich driverov máme na PC tzv. virtuálne COM porty, ktoré fungujú rovnako ako reálne sériové rozhranie.

Ak používame na komunikáciu USART, musíme zvoliť SPRÁVNU frekvenciu oscilátora. Dôvod je ten, že USART využíva systémovú frekvenciu k generovaniu prenosovej rýchlosti (Baud rates). Prichádzajúce data Rx a odchádzajúce data Tx majú definovanú frekvenciu. Procesor by mal mať zvolenú frekvenciu oscilátora tak, aby platilo:  $f_{osc} \text{ deleno } (8 \text{ alebo } 16) \text{ a deleno } BR \text{ bolo celé číslo bezo zvyšku.}$  Ako si neskôr ukážeme, prenosová rýchlosť vysielača a prijímača sa nesmie líšiť o viac ako 5%. My ale nespĺňame podmienku  $f_{osc}$ . My máme 16MHz.

USART vysiela a prijíma dátové rámce (data frames) ktoré sú tvorené z:

- 1 Start bit
- 5, 6, 7, 8 alebo 9 dátových bitov.
- Žiadnou, párnou, nepárnou... paritou. (**NO**,

## even, or odd parity bit)

- 1 alebo dva stop bity.

Je jedno aký formát zvolíme, vysielateľ Tx a prijímač Rx musia byť konfigurované rovnako.

Najčastejší formát je: 1S, 8data, NP, 1P.

Komunikáciu zabezpečujeme cez TTL-USB prevodník signálov.

**Aj toto je počítačová literatúra,  
literatúra faktu:**

### 2.5 Integrované obvody asynchronních vysílačů (UART)

Převod paralelních dat na sériová za účelem přenosu a opačný převod při příjmu zajišťují v portech COM k tomu určené integrované obvody UART (Universal Asynchronous Receiver-Transmitter – univerzální asynchronní přijímač a vysílač). Stejně integrované obvody generují a zpracovávají řídicí signály rozhraní. Porty COM počítačů IBM PC XT/AT jsou založeny na integrovaných obvodech, které jsou na úrovni registrů kompatibilní s řadou

UART 8250, 8250/16450/16550A. Tato řada je vylepšením původního modelu, což vedlo ke zvýšení provozní rychlosti a při intenzivní výměně dat se snížila spotřeba energie a zátěž procesoru. Všimněte si, že:

- ♦ 8250 se vyznačuje chybami návrhu (výskytem chybných přerušení), se kterými se počítalo v systému BIOS počítačů řady XT.
- ♦ 8250A: Některé chyby původního návrhu jsou opraveny, ale v důsledku toho se ztrácí kompatibilita se systémem BIOS řady XT. Tyto integrované obvody fungují v některých modelech řady AT, ale nelze je použít při rychlostech 9 600 b/s.
- ♦ 8250B: Chyby obvodů 8250 a 8250A jsou opraveny, kompatibilita se systémem BIOS řady XT je obnovena snížením původní chyby přerušení. Funguje u řady AT pod systémem DOS (s výjimkou rychlosti 9 600 b/s).

8250e. Tyto integrované obvody mají nižší provozní rychlosti s ohledem na požadavky ze systémové sběrnice. Neumožňují odkazování na své registry v následných cyklech sběrnice procesoru a je nutné skládat softwarové proslavy mezi odkazy procesoru, aby tyto integrované obvody správně fungovaly.

❗ počítačích třídy AT se používají následující upravené varianty integrovaných obvodů UART:

- ♦ 16450: Vysokorychlostní verze obvodů 8250 pro řadu AT. Nemá chyby obvodů 8250, neposkytuje plnou kompatibilitu se systémem BIOS řady XT.
- ♦ 16550: Vylepšená verze obvodu 16450. Umožňuje výměnu dat pomocí kanálu DMA. Obsahuje také vyrovnávací paměť FIFO, kterou nelze používat kvůli chybné funkci.
- ♦ 16550A: Má funkční 16bajtové vstupní a výstupní vyrovnávací paměti FIFO a podporu DMA. Jedná se o čip, který je vhodné použít v počítačích třídy AT při rychlostech výměny dat 9 600 b/s a vyšších. Většina integrovaných obvodů řadičů vstupně-výstupního portu v moderních čipových sadách je s tímto čipem kompatibilní.

Z hlediska programátora jsou integrované obvody 16550A UART sadou registrů. Přístup k nim je určen adresou (posun adresového registru vzhledem k adrese BASE portu) a hodnotou bitu DLAB (bit 7 z registru ICR). Ve vstupně-výstupním adresovém prostoru zabírá integrovaný obvod osm sousedních adres. Tabulka 2.4 uvádí registry UART 16550A a jejich přístupové metody. Integrované obvody 8250 se liší v tom, že nemají registr ICR a žádné funkce vyrovnávací paměti FIFO a přístupu DMA.

## Charakteristiky prenosových systémov

### Smer toku dát:

**Simplex, SX** (jednosmerne): dáta sú prenášané jedným smerom.

### Half-duplex, HDX

 (polovičný duplex):

Prenosové stanice sa vymieňajú pri posielaní dát. „Prepínač smeru vysielania“  
Pri prenose sa využíva tzv. HANDSHAKING (hardwareový a softwarový) – dohodnutý spôsob potvrdzovania prenosu.

### Full-duplex, FDX

 (plný duplex) (duplex - obojsmerná prevádzka):

Dáta môžu byť posielané oboma smermi súčasne.

Ako prenosové médium sa používajú dva páry vodičov.

### Rýchlosť prenosu dát:

Počet prenesených bitov za jednotku času [bit/s; bps]

Modulačná rýchlosť (Počet symbolov prenesených za jednotku času) – Baud rate.

[Baud - Bd].

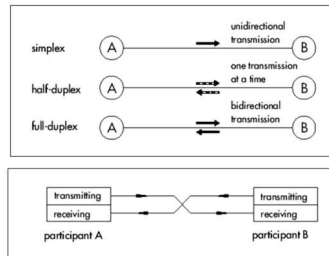
Baud rate – počet zmien úrovne signálu za sekundu.

Pre dvojtavovú moduláciu platí: Prenosová rýchlosť [bps]= Baud rate [Bd]

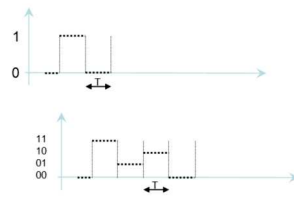
Štandardne používané rýchlosti sú:

( 50, 110, 300, 600, 1200)

2400, 4800, **9600**, 14400, 19200, 28800, 38400, 57600, 76800, [115200] Bd, bps.



A, B sú účastníci komunikácie



6

zabezpečujúci prepnutie smeru. Pri prenose sa využíva tzv. HANDSHAKING – Dohodnutý spôsob potvrdzovania prenosu.

- full-duplex (plný duplex) (duplex - obojsmerná prevádzka): Dáta môžu byť posielané oboma smermi súčasne. Ako prenosové médium sa používajú dva páry vodičov.

Pri komunikácii medzi dvomi bodmi (typu *Point-to-point*) vysielacie a prijímacie linky – vodiče môžu byť spojené pomocou dvoch oddelených liniek. Prijímacia linka jedného účastníka je prepojená s vysielacou linkou druhého účastníka. Komunikácia v takomto systéme, *Point to point* môže byť riadená softwarovo alebo pomocou riadiacich liniek.

Rýchlosť prenosu sa udáva počtom prenesených bitov za jednotku času [bit/s; bps]. Pri diaľkovom prenose dát sa používa modulovaný signál. Rýchlosť prenosu, sa vtedy vyjadruje tzv. modulačnou rýchlosťou - Baud rate. Jednotkou je Baud (Bd). Baud rate udáva počet zmien úrovne signálu za sekundu. Ak použijeme dvojtavovú moduláciu rýchlosť v baudoch bude zodpovedať rýchlosti v bitoch za sekundu. Ak n je počet stavov modulácie potom rýchlosť v bit/s=baud\*log<sub>2</sub>n .

Ešte jeden pokus o vysvetlenie: Prenosová rýchlosť  $\leftarrow, \rightarrow$  Modulačná rýchlosť. Ak počet stavov signálu odpovedá jednému bitu,

Najskôr treba povedať, že hovoríme o dvojbodovom spojení komunikujúcich zariadení. Ak komunikuje len dve zariadenia, netreba ich adresovať.

Sériový prenos je vhodný na prenos dát pri komunikácii medzi dvomi účastníkmi, ale aj medzi niekoľkými. Charakteristické vlastnosti komunikácie sú:

### smer toku dát,

V podstate poznáme tri rozdielne spôsoby komunikácie:

- Simplex (jednosmerne): dáta sú prenášané jedným smerom. Pri tomto prenose sa nedá zistiť či prenášané dáta boli prijaté.
- half-duplex (polovičný duplex): Prenosové stanice sa vymieňajú pri posielaní dát. Vyžaduje len jeden pár vodičov. Na každom konci musí byť prepínač,

prenosová rýchlosť sa rovná modulačnej rýchlosti. Modulácia je vlastne transformácia digitálnych signálov analógovým signálom.

1b – 2 stavy. 2b – 4stavy, ...

Ešte by sme mali spomenúť: Kapacitu prenosového kanálu.

Šírka pásma prenosového kanála je vlastne rozhas frekvencií, ktoré sa dajú kanálom preniesť bez rušenia a strát.

Už Nyquist dokázal, že ak máme spojitý signál s maximálnou frekvenciou  $f$ , tak ho možno charakterizovať počtom  $2f$  vzoriek za sek. A dá sa z týchto vzoriek zrekonštruovať. Shannon sa venoval tomu ako vplýva šum na prenosové vlastnosti kanálu.

**Kódovanie:** Priradenie binárneho čísla (byte, word) bližšie neurčeným spôsobom k znaku.

Morseho kód:

A	..-.	N	---.	0	-----
B	...-	O	---.	1	-----
C	..-.	P	...-	2	-----
D	..-.	Q	...-	3	-----
E	..	R	...-	4	-----
F	...-	S	...-	5	-----
G	...-	T	-	6	-----
H	....	U	...-	7	-----
I	..	Y	...-	8	-----
J	...-	W	...-	9	-----
K	...-	X	...-	.	----- (comma)
L	...-	Z	...-	,	----- (period)
M	...-			?	-----

„nula“  
„jednotka“  
„nič“

Figure 2. Subset of International Morse Code

Baudotov kód:

Mark = 1  
Space = 0

00	01	02	03	04	05	06	07	
NUL	E3	LF	A-	SP	S'	I8	U7	
08	09	0A	0B	0C	0D	0E	0F	
CR	D	R4	J	BEL	N,	F!	C:	K<
10	11	12	13	14	15	16	17	
T5	Z+	L>	W2	H&	Y6	P0	Q1	
18	19	1A	1B	1C	1D	1E	1F	
09	B?	G&	F!G\$	M.	X	U;	LTR\$	
Letters	Figures	Control Chars.						

Kód	Počet dátových bitov na znak
Baudot	5
Trans code	6
American Standard Code of Information in Interchanging (ASCII)	7
Extended Binary Coded Decimal Interchange Code (EBCDIC)	8

Binary Letters Figures

0001	A	7
1001	B	7
0110	C	7
0100	D	7
0001	E	5
0101	F	1
1010	G	6
1010	H	STOP
0010	I	8
0101	J	1
0111	K	1
1001	L	1
1100	M	1
01100	N	2
11000	O	9
1010	P	0
1011	Q	1
01010	R	6
00101	S	BELL
10000	T	5
0011	U	7
1110	Y	2
10011	W	2
1101	X	7
10101	Y	6
10001	Z	-
00000	n/a	n/a
01000	CR	CR
00100	LF	LF
00100	SP	SP
11111	LTR\$	LTR\$
11011	PR\$	PR\$

Figure 3. The Baudot Code Set

Pod znakom rozumieme: číslice, písmená a tzv. špeciálne znaky, napr. ?, !, §, CR, LF atď. Ak má znakom rozumieť aj počítač, treba ich zakódovať.

**Kódovanie:** Je vlastne priradenie ku konkrétnemu znaku z jednej množiny znak alebo postupnosť znakov z inej množiny. Je vynikajúce ak to priradenie je jednoznačné. Ak

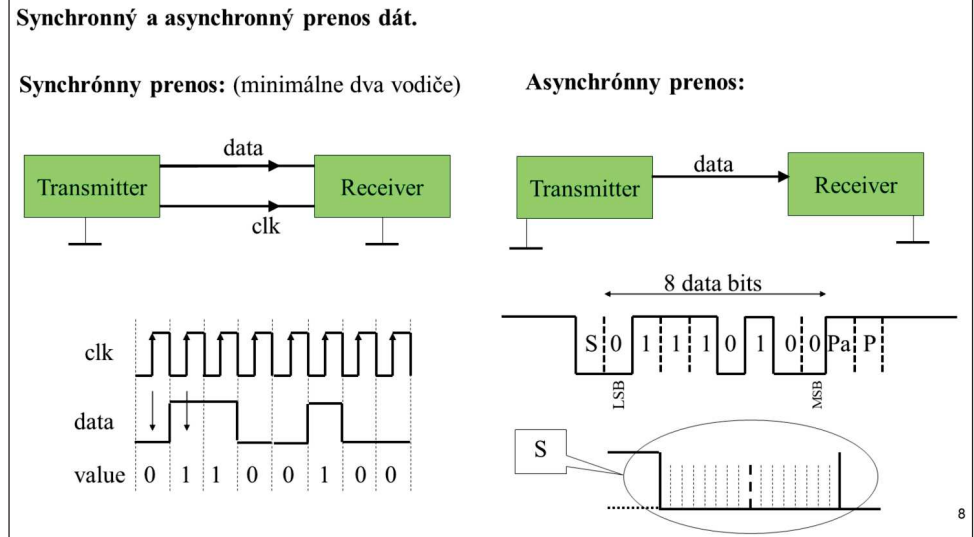
toto priradenie nie je známe, hovoríme o šifrovaní.

Jeden z najstaších spôsobov kodovania bola morseovka.

My sme sa už stretli s BCD kodovaním a ASCII kódom.

**Kód EBCDI - EBCDIC** (skratka pre Extended Binary Coded Decimal Interchange Code)

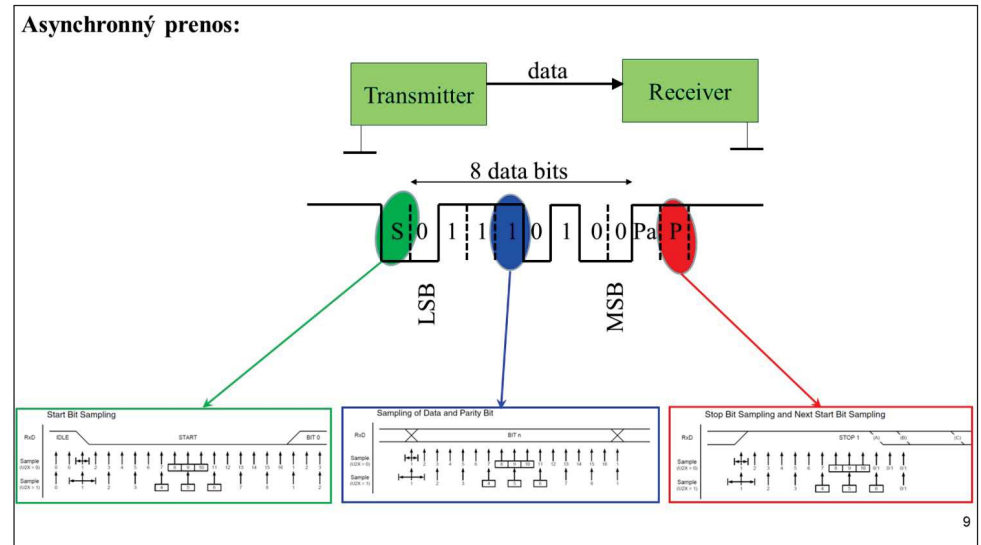
Rozšírený kód BCD na reprezentáciu číslíc, písmen a špeciálnych znakov. Tento kód potrebuje na reprezentáciu znaku jeden byte. EBCDIC vyvinula firma IBM a používa sa hlavne v minipočítačoch a strediskových počítačoch IBM.



**Synchronný prenos** informácie znamená, že na nejakom vodiči, alebo vodičoch sa nastaví úroveň a platnosť informácie sa potvrdí zmenou úrovne synchronizačného signálu, ktorý je platný pre obe strany - stanice. Synchronizačným impulzom sa teda informácie kvantujú. Synchronizačný impulz sa prenáša špeciálnym vodičom.

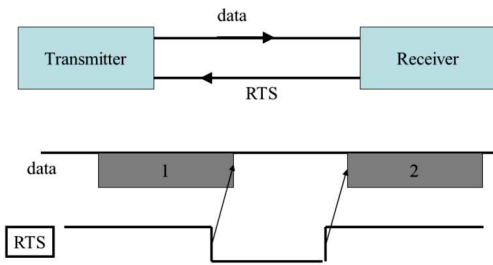
Pri **Asynchronnom prenose** sa neprenáša synchronizačný signál. Skutočnosť, že vysielač aj prijímač pracujú na tej istej frekvencii, ešte neznamená, že pracujú súfázne. S fázovanie sa deje na začiatku štartovacej sekvencie. Asynchronný prenos dát prenáša dáta v určitých sekvenciách. Kľudový stav pri asynchronom prenose je definovaný úrovňou log.1. Dáta sú prenášané dohodnutou rýchlosťou a začínajú štart bitom. Začiatok štart bitu synchronizuje prijímacie zariadenie. Vysielač aj prijímač obsahujú zdroj taktovacej frekvencie a synchronizačné

obvody. Prijímač považuje za štart bit taký prvý bit, kde v strede sa vyhodnotí log. 0. Nasledujúce bity sú vzorkované v strede intervalu. Po odvysielaní dátových bitov sa pripojí tzv. paritný bit a jeden resp. dva stop bity. Prenesené dátové bity sa považujú na strane prijímača za správne, ak má správnu hodnotu paritný aj stop bity.





**Handshaking** - podmienený prenos dát.  
 - potvrdzovanie pripravenosti komunikovať.  
 - potvrdzovanie prijatia dát, ukončenie prenosu ...  
 Hardware-ový,  
 Software-ový (XON (11h)/XOFF (13h)) ,  
 predpokladá duplexný prenos



Stĺpec	0	1	2	3	4	5	6	7
riadok								
0	NUL	DLE		0	@	P	·	p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	BS	CAN	(	8	H	X	h	x
9	HT	EM	)	9	I	Y	i	y
A	LF	SUB	*	:	J	Z	j	z
B	VT	ESC	+	;	K	[	k	{
C	FF	FS	,	<	L	\	l	
D	CR	GS	-	=	M	]	m	}
E	SO	RS	.	>	N	^	n	~
F	SI	US	/	?	O	_	o	DEL

### Detekcia chýb pri prenose. Poznámene 5 typov parity:

1. žiadna

2. **EVEN parity**

sum of all 1's must be even

data bits:	parity bit	Σ 1's
01101100	0	4
01101101	1	6

3. **ODD parity**

sum of all 1's must be odd

data bits:	parity bit	Σ 1's
01101100	1	5
01101101	0	5

#### Chyba komunikácie:

Okrem chyby parity sa môžu objaviť ďalšie chyby  
 Preplnenie vstupného buffra na prijímacej strane.  
 Chyba rámca (napr. stop bit sa neobjaví tam kde by sa mal)

Najjednoduchšie: - zopakuj chybný prenos - !!! Timeout !!!

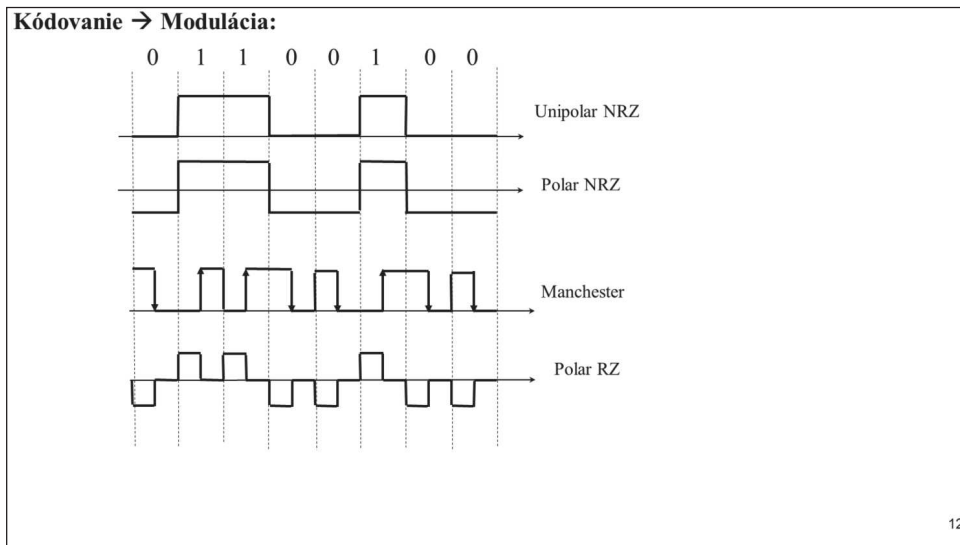
4. Mark = „1“  
 5. Space = „0“

Ak má komunikácia prebiehať korektne (synchronná alebo asynchronná), treba okrem prenosovej rýchlosti a zosynchronizovania zabezpečiť aj koordináciu na strane vysielača a prijímača. Prijímač musí potvrdiť pripravenosť prijať dáta zo strany vysielača. Signálom, napr.: RTS = 1 (Request To Send) oznamuje prijímač vysielaču, že je pripravený prijať ďalšie dáta. Ak sa prijímací buffer preplní signál RTS sa nastaví na nulu a vysielač pozastaví vysielenie dát.

Hardwarové potvrdzovanie pripravenosti prijať správu možno použiť pre systémy typu: *Point to point*. Ak zapojíme potvrdzovacie vodiče pomocou tzv. *drôtového OR* resp. *AND* môžeme hardwarový handshaking použiť aj pre zapojenie *multi point*.

Je jedno či komunikujeme synchronne alebo asynchronne, s alebo bez potvrdzovania pripravenosti prijatia a spracovania dát, vždy môže nastať chyba pri prenose. Na detekovanie chýb bolo vypracovaných viacero techník: Najjednoduchšie je chybu odstrániť. To ale vyžaduje zložité kódovanie – bity navyše. Ako najjednoduchšie sa javí spôsob, pri ktorom sa chybný prenos jednoducho zapakuje. Jeden zo spôsobov detekcie chyby pri prenose jedného znaku je kontrola parity. Nezapadni spomenúť „**timeout**“.

Kontrola prenosu metodikou: Párna (EVEN) parita, predpokladá páry počet prenesených jednotiek (vrátane paritného bitu). Metodika nepárnej parity predpokladá nepárny počet jednotiek. Táto metodika dokáže zachytiť len jednu chybu.



Logickým úrovniam priradíme napätie. Je zrejmé, že unipolárny signál (binárne data) majú jednosmernú zložku. Čo nie je vhodné na prenos na väčšie vzdialenosti.

Zakódovaný signál treba treba previesť na nejakú fyzikálnu veličinu, na jej zmeny. Tomuto sa hovorí Modulácia.

My sa tu budeme venovať len elektrickému médium (iné, napr. optika).

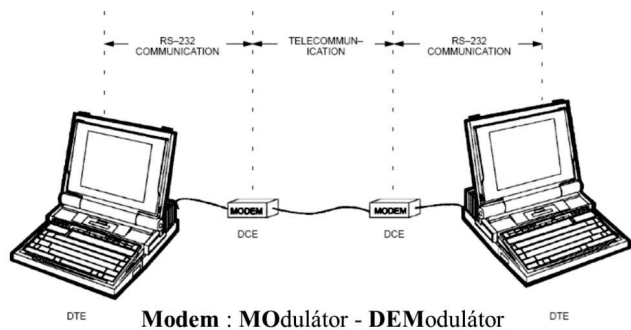
Na začiatku sa často používala tzv. prúdová slučka (10-ky mA. Výhoda použiteľný na veľké vzdialenosti.

Napät'ový signál je najčastejšie používaný, ale má nevýhodu v tom, že je ľahko zarušiteľný. Podľa toho čo na elektrickom signále meníme hovoríme o amplitúdovej, frekvenčnej, resp. fázovej modulácii. Prvý spôsob je priamy:

### RS 232 ( x ) alebo V.24 interface

Základné pojmy definované touto normou:

- **DTE** (Data Terminal Equipment - *Koncové dátové zariadenie*) je počítač, resp. terminál a
- **DCE** (Data Communications Equipment – *Zariadenie dátovej komunikácie*) napríklad modem, zapisovač.



13

Rôzne metodiky kódovania, napr.: NRZ, Manchester, atď. Definujú spôsob akým sú reprezentované binárne stavy pri prenose sériových dát. V priebehu rokov bolo vypracovaných viacero noriem, ktoré definujú rôzne úrovne, frekvencie, prenosové rýchlosti, dĺžku vodičov, ... . Niektoré časom zanikli, a niektoré po viacnásobnom prepracovaní sa zachovali do dnešných dní:

Základné pojmy definované normou RS232 resp. V.24 sú:

- DCE (Data Communications Equipment) napríklad modem, zapisovač, ... . a
- DTE (Data Terminal Equipment) je počítač, resp. terminál.

**Modem** je zloženina slov MOdulátor a DEModulátor a je

to zariadenie, ktoré konvertuje číselnú informáciu z PC na zvukové signály (šírka telefónneho pásma je od 300 Hz do 3400 Hz), tie potom pošle po telefónnej linke inému modemu, ktorý spätne vykoná prevod analógového signálu prijatého z linky na digitálne impulzy.

Prepojenie dvoch zariadení typu: Point-to-point obvykle používa RS 232 interface. Kompletný popis tohoto interface pre štvorvodičový full-duplex prenos, ako aj definovanie funkcií potvrdzovania prenosu sú dané americkou normou RS232 alebo identickou medzinárodnou normou V.24.

## RS232 – V.24

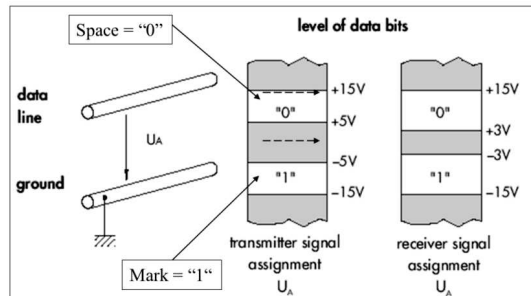
Prenos dátových a riadiacich signálov:

- dáta v negatívnej logike  
(0: high; 1: low)

- riadiace signály v pozitívnej logike  
(1: high; 0: low)

Iné názvy: signál log. 0 - "Space" a  
signál log. 1: - "Mark".

data	Control signal	level	Voltage range
Log. 0	Log. 1	high	+3 to +15V
Log. 1	Log. 0	low	-3 to -15V

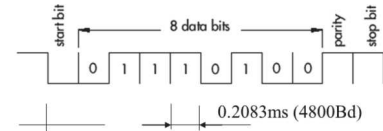


Obmedzenie podľa RS232 F:  
 $du/dt = 30V/\mu s$ ;  
Dĺžka vedenia: < 2500 pF

14

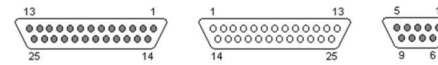
Z fólie je zrejmé, že napätové úrovne pre dátové bity a riadiace bity sú opačné. Rozsah napätí pre logické úrovne je prísnejšie určený ako na strane prijímača.

## USART - RS232



Vysielač a prijímač musia byť konfigurované rovnako – musia mať rovnaké konfiguračné parametre:

- Prenosová rýchlosť - Baud rate (medzi 50 až 19.2 kbit/s),
- parita (bezparity, párna alebo nepárna parita) a
- počet stop bitov (1, 1.5 alebo 2).



Vidlica (DTE) a zásuvka (DCE) 25 pinov

Received Data a Transmitted Data (RxD, TxD)  
S, b0, b1, b2, ... (PP, NP, BP), P (1.5P, 2P)

Request to Send (RTS): DTE chce vysielať  
Clear to Send (CTS): DCE môže prijať a ďalej posla.

Data Set Ready (DSR): DCE je pripojený k linke.  
Data Carrier Detect (DCD): DCE zachytilo nosnú frekv.  
Signál OK.

Data Terminal Ready (DTR): „DTE (počítač) je pripravený“

Ring Indicator (RI): DCE indikuje vyzváňací signál.

GND Cable Shield (GND): Tienenie, ochranná zem

15

Komunikácia predpokladá, že USART a RS232 sú správne nakonfigurované/zapojené na oboch komunikujúcich stranách. Predpokladáme asynchrónnu komunikáciu.

Vysielač a prijímač musia byť konfigurované rovnako – musia mať rovnaké konfiguračné parametre:

- Prenosová rýchlosť - Baud rate (medzi 50 až 19.2 kbit),
- parita (bezparity, párna alebo nepárna parita) a
- počet stop bitov (1, 1.5 alebo 2).

Ako konektor pre toto rozhranie (RS232) sa používa 25 pinový konektor typu D sub (Cannon), pričom na DTE (počítač) je vidlica („samček“), modem má zásuvku.

Seriové porty sú v dvoch prevedeniach – rozmeroch: Umisťujú sa na zadnú stranu počítača:

25 kolíkový konektor. Niekedy sa nazýva aj tzv.

Veľký variant štandardu rozhrania RS232.

9 kolíkový konektor . Niekedy sa nazýva aj tzv. stredný variant.

Na overovanie správnosti príjmu sa používa menej signálov: RTS, CTS , DCD , DTR .

RS232 – V.24

Cannon		Názov	CCITT V.24	Smer počítač – modem	Popis a funkcia
DB25	DB9				
1		GND	101	—	Cable Shield, Tienenie, ochranná zem
2	3	TXD	103	→	Transmitted Data, Vysielané údaje.
3	2	RXD	104	←	Received Data, Prijímané údaje
4	7	RTS	105	→	Request to Send, Výzva na vysielanie údajov
5	8	CTS	106	←	Clear to Send DCE, pripravený prijať a odvyslať data
6	6	DSR	107	←	Data Set Ready, DCE je pripravený uskutočniť spojenie
7	5	GND	102	—	Signal Ground, Signálová zem
8	1	DCD	109	←	Data Carrier Detect, Úroveň signálu je v predpísaných medziach
9,10		-	-	-	Rezervované
11		STF	126	→	Select Transmitt Channel, Voľba vysielacej modulačnej frekvencie
12		S.CD	122	←	Secondary Carrier Detect, Ako DCD, pre sekundárny kanál
13		S.CTS	121	←	Secondary Clear to Send, Ako CTS, pre sekundárny kanál
14		S.TXD	118	→	Secondary Transmitted Data, Ako TxD, pre sekundárny kanál
15		TCK	114	←	Transmission Signal Element Timing, Časovanie prvkov vysielaného signálu
16		S.RXD	119	←	Secondary Received Data, Ako RxD, pre sekundárny kanál
17		RCK	115	←	Receiver Signal Element Timing, Časovanie prvkov prijímaného signálu
18		LL	141	→	Local Loop Control, Riadenie lokálnej spätnéj slučky
19		S.RTS	120	→	Secondary Request to Send, Ako RxD, pre sekundárny kanál
20	4	DTR	108	→	Data Terminal Ready, DTE (počítač) je pripravený
21		RL	140	→	Remote Loop Control, Riadenie vzdialenej spätnéj slučky
22	9	RI	125	←	Ring Indicator, DCE (modem) hlási, že prijíma volací signál
23		DSR	111	→	Data Signal Rate Selector, Voľba prenosovej rýchlosti
24		XCK	113	→	Transmit Signal Element Timing, Časovanie prvkov vysielaného signálu
25		TI	142	←	Test Indicator, Signalizácia poruchy

16

Označenie signálov a ich rozloženie na 25-pinovom konektore je v tabuľke. Norma definuje komunikáciu cez tzv. primárny aj sekundárny kanál, vrátane plného handshakingu (potvrdzovania). Takýto systém však používa málokto a preto sa prakticky vždy používa len nejaká podmnožina týchto signálov. Niektoré aplikácie dokonca vystačia len s tromi z nich (TxD, RxD a GND). Význam deviatich najpoužívanejších signálov tohoto rozhrania (použitých na 9-pinovom konektore) je nasledovný: **Received Data a Transmitted Data (RxD, TxD)** je vstup (výstup) údajov v sériovom tvare. Ten sa začína štart bitom, pokračuje 5 až 8 datovými bitmi, pričom najmenej významný bit ide prvý. Pri prenose s kontrolou parity nasleduje paritný bit (párny, nepárny) a na záver jeden, jeden a pol alebo dva stop bity. Signál TxD generuje DTE a prijíma DCE. V pokojovom stave (DCD nie je nastavené)

má byť na výstupe RxD logická jednotka.

**Request to Send (RTS):** Ak je počítač (DTE) pripravený vyslať dáta periférii (DCE), signál RTS je nastaví do jednotky a periféria sa musí pripraviť na vysielanie. V poloduplexnej prevádzke sa zároveň zablokuje príjem. Keď je všetko v poriadku, DCE nastaví signál CTS a dá tak najavo pripravenosť.

**Clear to Send (CTS):** Týmto signálom dáva DCE najavo pripravenosť na vysielanie. Keď mu ich začne DTE posielat', jeho úroveň sa znova zmení.

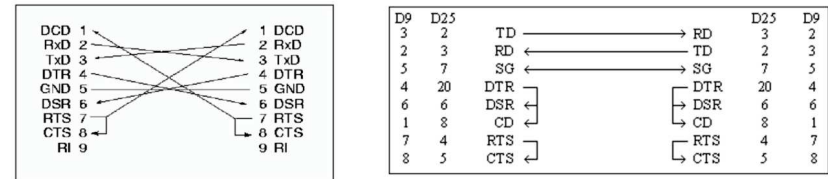
**Data Set Ready (DSR):** Aj tento signál nastavuje DCE a oznamuje tak, že je pripojený k telekomunikačnej linke.

**Data Carrier Detect (DCD):** Týmto signálom DCE indikuje, že prijíma nosný signál (*carrier*) zo vzdialeného zariadenia v požadovanej kvalite.

**Data Terminal Ready (DTR):** DTR indikuje pripravenosť počítača (DTE) prijímať dáta. Aby mohol modem (DCE) nastaviť signál DSR, musí najprv byť nastavený tento signál.

**Ring Indicator (RI):** Ak sa využíva aj tento signál, potom indikuje prítomnosť vyzváacieho signálu na strane modemu (DCE).

### Nulový modem (DTE - DTE) priame prepojenie počítačov

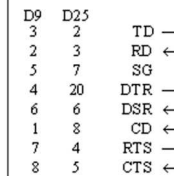


Tri vodiče: TxD, RxD, GND

Data Terminal Ready  
Data Set Ready  
Data Carrier Detect

Request To Send  
Clear To Send

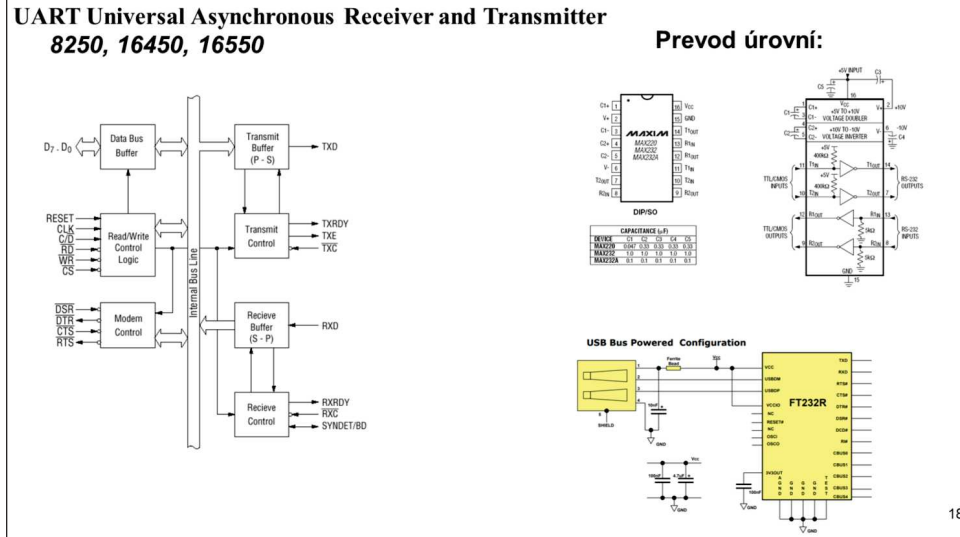
#### LoopBack Plug



„Slučka“: PC ← → PC

17

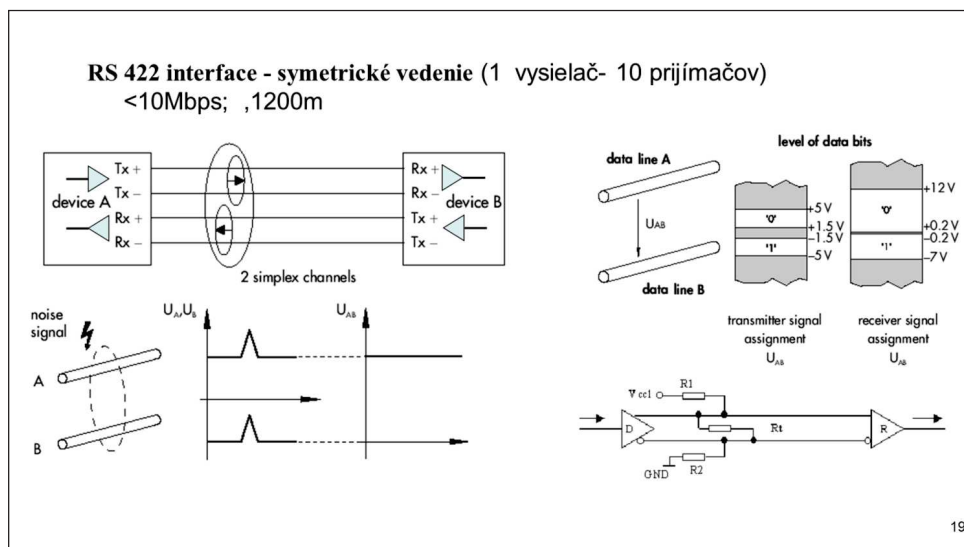
Všetky signály využíva len málokto. Preto sa prakticky vždy používa len nejaká podmnožina týchto signálov. Niektoré aplikácie dokonca vystačia len s tromi z nich (TxD, RxD a GND). Význam deviatich najpoužívanejších signálov tohoto rozhrania (použitých na 9-pinovom konektore) je nasledovný:



prerušená. Príkladom je obvod 8251 pripojiteľný k 8-bitovým mikropočítačom, alebo 8250A, príp. 16550A používaný v počítačoch IBM PC. Najmodernejšie obvody okrem obvodu UART obsahujú aj Plug and Play rozhranie pre Windows, napríklad TL16PNP550A (Texas Instruments).

Realizáciu tohoto rozhrania nám uľahčujú integrované obvody, ktoré prevádzajú signály z a do úrovni TTL. Najstaršie obvody SN 75150/154 (Texas Instruments) a MC 1488/1489 (Motorola) vyžadovali napájanie  $\pm 12$  V, modernejšie MAX 232 (Maxim), LT1180A (Linear Technology) už obsahujú príslušný menič a vystačia s jediným napájaním +5 V. Najmodernejšie obvody obsahujú celé rozhranie s príslušným počtom vstupov a výstupov pre 9-pinový konektor, napr. SN 75LV4735 (Texas Instruments).

Na tomto mieste sa zmienime ešte o obvodoch, ktoré realizujú prevod dát z paralelného tvaru na sériový. Sú to tzv. **obvody UART** (Universal Asynchronous Receiver and Transmitter). Takýto obvod generuje štart bit, stop bit, prevádza byty na sériovú postupnosť bitov. Okrem toho obvykle obsahuje aj vyrovnávaciu pamäť a generuje



obrázkom. Parametre vysielateľa musia byť garantované aj pre záťaž 54 ohmov. Táto záťaž je daná zakončovacími odporami na koncoch vedenia.

Interface RS 422 je vhodný najmä na rýchle sériové prenosy a na veľké vzdialenosti. RS422 umožňuje maximálne pripojiť paralelne 10 prijímačov na jeden vysielateľ. Na krátke vzdialenosti možno komunikovať prenosovou rýchlosťou 10Mbit/s. Prenosová rýchlosť sa s dĺžkou vedenia znižuje: dĺžka vedenia do 1200m – prenosová rýchlosť sa obmedzí na 100kbit/s.

RS 422 možno realizovať ako 4-vodičové zapojenie: full-duplex, resp. ako dvojvodičové: simplex.

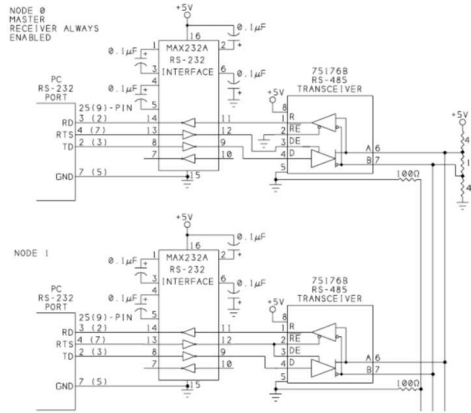
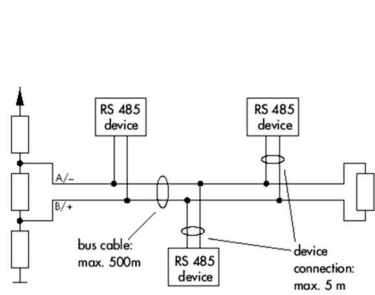
Pri zapojovaní výstupu vysielateľa (Tx) so vstupom prijímača (Rx) treba dodržať správnu polaritu signálov:

Interface RS422 prenáša informáciu – logické úrovne ako diferenčné napätie medzi dvomi vodičmi A a B. T.j. aj keď sa na tieto vodiče naindukujú rušivé signály, diferenčné napätie zostane to isté:

Napäťové úrovne pre vysielateľ a prijímač sú dané



**RS 485 interface až 32 zariadení  
9600 bps – 1,2km**



20

Toto rozhranie vzniklo pôvodne ako rozšírenie RS-422-A, pričom prevzalo signálové úrovne a doplnila sa možnosť poloduplexnej zbernicovej komunikácie. Norma je dosť stručná, neobsahuje napríklad definíciu konektorov, maximálnu dĺžku vedenia, rýchlosť a pod. Toto rozhranie je použité aj v systéme Profibus.

Na prenos sa používa symetrické vedenie, prenosová vzdialenosť je typicky až do 1,2 km, ale len rýchlosťou 9600 bps. Maximálnou rýchlosťou 12 kbps však môžeme komunikovať len na vzdialenosť 100 m. Na jedno vedenie môžeme pripojiť až 32 zariadení, z ktorých každé môže prijímať i vysielať data.

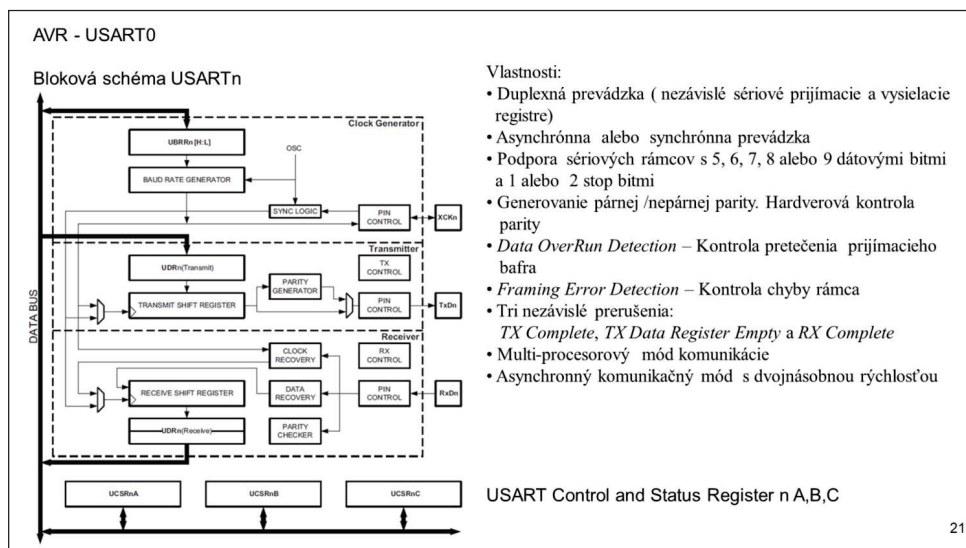
Ako vodiče pre toto rozhranie sa používa bežná krútená dvojlínka, pre náročnejšie aplikácie tienená. Pre priemyselné použitie (Profibus) je s dvojitém tienením a vonkajším obalom z PVC, PE alebo PUR, ktorý kábel

chráni pred nepriaznivými vplyvmi.

Typický kábel má menovitú impedanciu  $150 \Omega$  ( $135 - 165$ ) pre  $3 - 20$  MHz, s útlmom menším ako  $40$  dB/km, odporom  $\leq 110 \Omega/\text{km}$  a kapacitou  $\leq 30$  pF/m. Minimálny prierez vodičov je  $0,34 \text{ mm}^2$ , max. dĺžka odbočky  $0,3$  m. Každý segment má byť ukončený terminátorom. Podľa normy Profibus má okrem ukončovacieho odporu  $220 \Omega$  obsahovať aj dva odpory  $390 \Omega$  pripojené na napájanie, ktoré definujú stav zbernice aj vtedy, keď žiadna stanica nevysiela. Preto musí byť v systéme Profibus terminátor napájaný.

Napät'ové úrovne sú definované podľa normy V.11, teda rovnaké ako pri RS-422-A. Budiče obvodov sú napr. 75174/175, kombinovaný SN 75156A, moderný SN 75ALS176B, s ochranami vstupov do nepriaznivých prostredí SN 75LBC184 (Texas Instruments).

Ak používa niekoľko účastníkov jedno prepojenie, protokol musí zabezpečiť, že v danom čase je aktívny len jeden vysielač. Ostatné vysielače v tom čase musia uvoľniť linku a prepnúť výstupy do vysokej impedancie.



synchronizačnému a bloku obnovy dát. Blok obnovy dát sa používa len pri asynchrónnom prijíme. Súčasťou bloku obnovy dát je aj obvod kontroly parity, riadiaca logika, posuvný register a dvojúrovňový prijímací bafer (**UDR**). Prijímač podporuje tie isté formáty rámcov ako vysielač a môže detekovať chybu rámca, pretečenie dát a chybu parity.

USART je tvorený tromi blokmi: generátor hodín (*Clock Generator*), vysielač (*Transmitter*) a prijímač (*Receiver*). Riadiace registre sú spoločné.

Logika generovania hodinových signálov je vytvorená zo synchronizačnej logiky externého hodinového signálu použitého pri synchronnej prevádzke typu **SLAVE** a generátora prenosovej rýchlosti. Pin **XCK** (*Transfer Clock*) je použitý len pri synchronnom prenosovom móde.

**Vysielač** pozostáva zo zapisovacieho bafra, sériového posuvného registra, generátora parity a riadiacej logiky pre tvorbu rôznych sériových formátov rámcov. Zápisový bafer umožňuje spojitý prenos dát bez oneskorenia medzi rámcami.

**Prijímač** je najzložitejšia časť **USART**-u vďaka bloku

**Generátor prenosovej rýchlosti**

- Normal Asynchronous,
- Double Speed Asynchronous,
- (Master Synchronous a Slave Synchronous mode).

**Equations for Calculating Baud Rate Register Setting**

Operating Mode	Equation for Calculating Baud Rate <sup>(1)</sup>	Equation for Calculating UBRRn Value
Asynchronous Normal mode (U2Xn = 0)	$BAUD = \frac{f_{osc}}{16(UBRRn + 1)}$	$UBRRn = \frac{f_{osc}}{16BAUD} - 1$
Asynchronous Double Speed mode (U2Xn = 1)	$BAUD = \frac{f_{osc}}{8(UBRRn + 1)}$	$UBRRn = \frac{f_{osc}}{8BAUD} - 1$
Synchronous Master mode	$BAUD = \frac{f_{osc}}{2(UBRRn + 1)}$	$UBRRn = \frac{f_{osc}}{2BAUD} - 1$

BAUD - Baud rate (v bitoch za sekundu, [bps])  
 $f_{osc}$  - frekvencia systémového oscilátora v Hz.  
**UBRR** - Obsahy registrov **UBRRH** a **UBRRL**, (0 až 4095) – 16bitov.  
 Niekoľko príkladov výpočtu hodnôt UBRR vid'. nasledujúca tab.

22

**Generátor prenosovej rýchlosti**

Logika generovania taktovacieho signálu vytvára taktovacie signály pre vysieláč aj prijímač. **USART** podporuje štyri módy generovania taktovacieho signálu: *Normal Asynchronous*, *Double Speed Asynchronous*, *(Master Synchronous a Slave Synchronous mode)*.

Bit **UMSELn** v **USART Control and Status Register C (UCSRnC)** prepína medzi asynchrónnou a synchrónnou prevádzkou. *Double Speed* (len asynchrónny mód) je riadená bitom **U2Xn**. Tento sa nachádza v registri **UCSRnA**.

Pri popise ignorujeme synchrónnu prevádzku, t.j. (**UMSELn** = 0).

Popis signálov:

- txclk** *Transmitter clock* (Vnútorný signál).
- rxclk** *Receiver base clock* (Vnútorný signál).
- xcki** *Input from XCK pin* (Vnútorný signál). Použité len pri synchrónnom **SLAVE** móde.
- xcko** *Clock output to XCK pin* (Vnútorný signál). Použité len pri synchrónnom **MASTER** móde.
- fosc** *XTAL pin frequency* (Systémové hodiny).

Generátor vnútorných hodinových signálov je použitý aj pre asynchrónnu prevádzku aj pre synchrónny Master mód. Register prenosovej rýchlosti (*USART Baud Rate Register - UBRRn*) a počítadlo smerom nadol pracujú vo funkcii programovateľnej preddeličky – generátora prenosovej rýchlosti.

Počítadlo smerom nadol, načítava systémové hodiny ( $f_{osc}$ ), predplní sa hodnotou registra **UBRRn** pokaždé, keď počítadlo dopočíta na nulu alebo keď sa zapíše do registra **UBRRL**. Taktovací signál sa generuje pokaždé, keď počítadlo dosiahne nulu. Tieto hodiny sú výstupom generátora taktovacej frekvencie prenosovej rýchlosti (=  $f_{osc}/(UBRRn+1)$ ). Vysieláč delí tento signál 2, 8 alebo 16 v závislosti od nastaveného módu. Výstup generátora prenosovej rýchlosti je použitý priamo v prijímači. Avšak prijímač, používa stavový automat, ktorý používa 2, 8 alebo 16 stavov v závislosti od módu nastaveného v **UMSELn**, **U2Xn** a **DDR\_XCKn** bitoch.

Examples of UBRRn Settings for Commonly Used Oscillator Frequencies (Continued)

Baud Rate (bps)	$f_{osc} = 16.0000$ MHz				$f_{osc} = 18.4320$ MHz				$f_{osc} = 20.0000$ MHz			
	U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1		U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	416	-0.1%	832	0.0%	479	0.0%	959	0.0%	520	0.0%	1041	0.0%
4800	207	0.2%	416	-0.1%	239	0.0%	479	0.0%	259	0.2%	520	0.0%
9600	103	0.2%	207	0.2%	119	0.0%	239	0.0%	129	0.2%	259	0.2%
14.4k	68	0.6%	138	-0.1%	79	0.0%	159	0.0%	86	-0.2%	173	-0.2%
19.2k	51	0.2%	103	0.2%	59	0.0%	119	0.0%	64	0.2%	129	0.2%
28.8k	34	-0.8%	68	0.6%	39	0.0%	79	0.0%	42	0.9%	86	-0.2%
38.4k	25	0.2%	51	0.2%	29	0.0%	59	0.0%	32	-1.4%	64	0.2%
57.6k	16	2.1%	34	-0.8%	19	0.0%	39	0.0%	21	-1.4%	42	0.9%
76.8k	12	0.2%	25	0.2%	14	0.0%	29	0.0%	15	1.7%	32	-1.4%
115.2k	8	-3.5%	16	2.1%	9	0.0%	19	0.0%	10	-1.4%	21	-1.4%
230.4k	3	8.5%	8	-3.5%	4	0.0%	9	0.0%	4	8.5%	10	-1.4%
250k	3	0.0%	7	0.0%	4	-7.8%	8	2.4%	4	0.0%	9	0.0%
0.5M	1	0.0%	3	0.0%	-	-	4	-7.8%	-	-	4	0.0%
1M	0	0.0%	1	0.0%	-	-	-	-	-	-	-	-
Max. (1)	1 Mbps		2 Mbps		1.152 Mbps		2.304 Mbps		1.25 Mbps		2.5 Mbps	

1. UBRRn = 0, Error = 0.0%

$$UBRRn = \frac{f_{osc}}{16 * BR} - 1$$

$1\ 000\ 000 / 9600 = 104$	$104.16$
$1\ 000\ 000 / 19200 = 52$	$52.083$
$1\ 000\ 000 / 115200 = 8$	$8.6805$
$\frac{f_{osc}}{16}$	BR
	? -1 ?

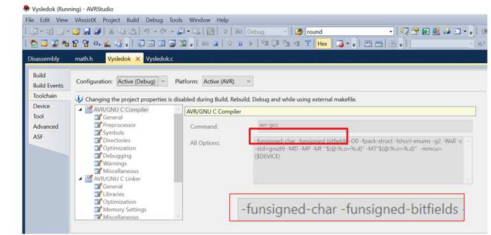
23

## Celočíselné delenie:

```
signed char klcis1 = 1;
signed char klcis2 = 2;
signed char klcis3 = 3;
signed char zacis1 = -1;
signed char zacis2 = -2;
signed char zacis3 = -3;
signed char Vysledok_d = 0;
```

```
Vysledok_d = klcis1/2; // 0b0000 0001 / 2 = 0b0000 0000 (1/2 = 0)
Vysledok_d = klcis2/2; // 0b0000 0010 / 2 = 0b0000 0001 (2/2 = 1)
Vysledok_d = klcis3/2; // 0b0000 0011 / 2 = 0b0000 0001 (3/2 = 1)
```

```
Vysledok_d = zacis1/2; // 0b1111 1111 / 2 = 0b0000 0000 (-1/2 = 0)
Vysledok_d = zacis2/2; // 0b1111 1110 / 2 = 0b1111 1111 (-2/2 = -1)
Vysledok_d = zacis3/2; // 0b1111 1101 / 2 = 0b1111 1101 (-3/2 = -1)
```



24

Celočíselné delenie: výsledky sú približné.

$$1\ 000\ 000 / 9600 = 104$$

$$1\ 000\ 000 / 19200 = 52$$

$$1\ 000\ 000 / 115200 = 8$$

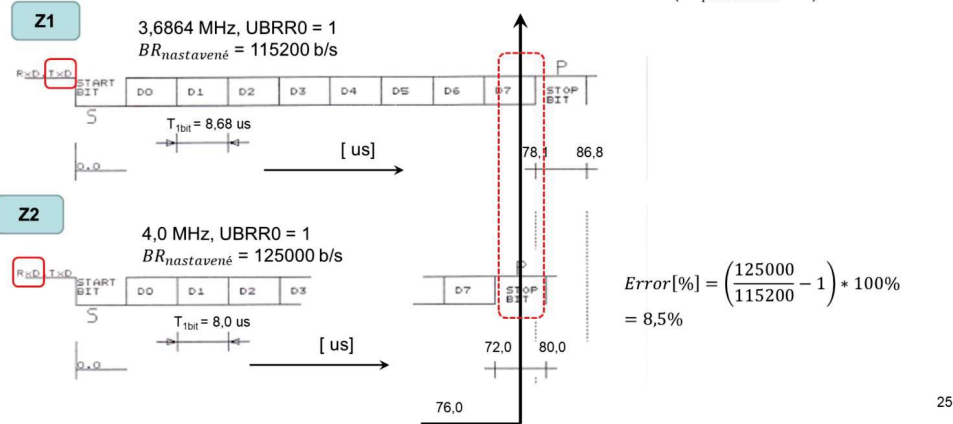
8b-vý microcontroller 80C552 bol stavaný na  $f_{osc} = 16$ MHz, ale bežne bol osadzovaný s  $f_{osc} = 11,0592$ MHz.

Prečo?

! 115200 b/s !

$$UBRR0 = \frac{f_{osc}}{16 * BR} - 1 \quad BR = \frac{f_{osc}}{16 * (UBRR0 + 1)}$$

$$Error[\%] = \left( \frac{BR_{nastavené}}{BR_{požadované}} - 1 \right) * 100\%$$



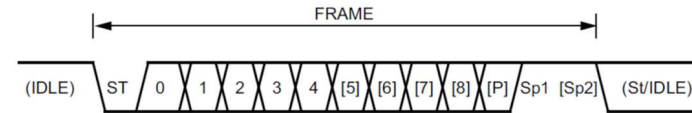
25

Máme dve zariadenia, ktoré majú komunikovať na BR = 115200 b/s. Jedno má  $f_{osc} = 3,6864$  MHz (1/8/4) a druhé  $f_{osc} = 4,0$  MHz (1/8/4). Zariadenie Z1 vysiela a Z2 prijíma. Z2 prijíma ako stoP bit, to čo Z1 odošle ako d7. Chyba je 8,5%. To je viac ako teoretických 5%.

#### Formát prenosových rámcov

Sériový rámec je prenášaný ako jeden znak, ku ktorému sú pripojené Start a stoP bit a ak treba tak aj paritný bit. USART podporuje všetky platné kombinácie (30) vytvorené z:

- 1 start bitu
- 5, 6, 7, 8 alebo 9 dátových bitov
- žiadneho, párneho alebo nepárneho paritného bitu
- 1 alebo 2 stop bitov (Pôvodne bolo možné v USART-och nastaviť aj 1.5 bitu.)



26

Rámec začína Start bitom, za ktorým nasleduje najmenej významný dátový bit. Potom nasledujú ďalšie dátové bity, až po maximálne deviaty. Najvýznamnejší dátový bit ako posledný. Ak je povolený paritný bit, vkladá sa za dátové bity, pred stoP bity. Keď je prenesený celý rámec, môže sa prenášať ďalší, alebo sa komunikačná linka nastaví do stavu pokoja – vysoká úroveň – log. 1.

Kombinácie formátov rámcov sú zrejmé z obrázku. Bity v zátvorke sú voliteľné.

**St** - Start bit, vždy nízka úroveň.

**(n)** - Dátové bity (0 až 8).

**P** - Paritný bit. Môže byť nastavený na párnou alebo nepárnou paritu.

**Sp** - StoP bit, vždy vysoká úroveň.

**IDLE** - Žiadny prenos cez komunikačné linky (**RxD** alebo **TxD**). Linka v **IDLE** móde musí byť na vysokej úrovni – v pokoji. Niekedy sa idle prekladá ako „voľnobeh“.

Formát prenášaného rámca **USART**-om sa nastavuje pomocou bitov: **UCSZn2:0**, **UPMn1:0** a **USBSn** v registroch **UCSRnB** a **UCSRnC**. Vysielač a prijímač sú nastavené rovnako.

Poznámka: Zmena ktoréhokoľvek bitu spôsobí zrútenie prebiehajúcej komunikácie aj v prijímači aj vo vysielači.

Počet dátových bitov v prenášanom rámci je daný nastavením bitov - *USART Character Size* (**UCSZn2:0**). Mód parity sa zapína a nastavuje pomocou bitov *USART Parity mode* (**UPMn1:0**).

Počet stop bitov v prenášanom rámci sa nastavuje bitom *USART Stop Bit Select* (**USBSn**). Prijímač ignoruje druhý stop bit. **FE** (*Frame Error*) sa nastaví len v prípade ak je prvý stop bit nulový.

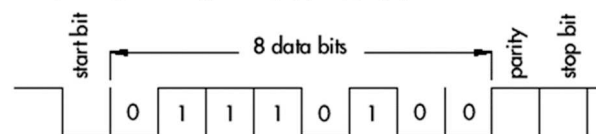
#### Výpočet paritného bitu

$$P_{even} = d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0$$

$$P_{odd} = d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1$$

Úloha: Vysielač odoslal nasledovný rámec.

- Napište hex-a hodnotu vysielaného bytu.
- Uveďte typ nastavenej parity.
- Uveďte akým spôsobom vyhodnocuje hardver prijímača správnosť paritného bitu, ak sa to deje priebežne. Po príchode paritného bitu vie vyhodnocovacia logika či bol prijatý správny paritný bit.
- Zapište algoritmus výpočtu chyby parity pri prijme: UCSR0A.UPE0.



27

Paritný bit je počítaný pomocou operácie exclusive-or zo všetkých dátových bitov. Ak použijeme nepárnu paritu, výsledok je **invertovaný**.

$P_{even}$  Paritný bit použitý pri párnej parite

$P_{odd}$  Paritný bit použitý pri nepárnej parite

$d_n$  n-tý dátový bit

Ak pri prenose použijeme paritný bit, umiestňujeme ho za dátové bity a pred prvý stop bit.

Inicializácia - USART0

```
#define F_CPU 16000000UL
#include <avr/io.h>
#define BAUD 115200
```

$$UBRR0 = \frac{f_{osc}}{16 * BR} - 1$$

0. #define MYUBRR F\_CPU/16/BAUD-1 Takto to platí len niekedy. Kedy?  
Ako to má byť zapísané správne?

```
// Inicializacia UARTu
void ini_USART0(unsigned int mybr){
  UBRR0 = mybr; // Set baud rate UARTu
  set_bit(UCSR0B, TXEN0); // // Enable Tx
  // Set frame format: 8b data, 2stoP bity
  UCSRC = (1<< UCSZ50)|(1<< UCSZ00)|(1<< USBS0);
}
```

1. #define MYUBRR F\_CPU / 16UL / BAUD - 1  
2. #define MYUBRR (((F\_CPU / (BAUD \* 16UL))) - 1)

3. #define MYUBRR (10 \* F\_CPU / 16 / BAUD-1) / 10  
4. #define MYUBRR ((F\_CPU + BAUD \* 8UL) / (BAUD \* 16UL) - 1UL)

28

USART treba inicializovať pred prvým použitím.

Inicializácia je tvorená:

- Nastavením prenosovej rýchlosti,
- nastavením parametrov rámca a
- povolenia vysielania alebo príjmu podľa potreby.

Ak chceme využívať prerušovací podsystem, musíme mať pri inicializácii zakázané globálne prerušenie. Predtým ako urobíme preinicializovanie, treba sa presvedčiť, že sa nevykonávajú žiadne prenosy. Bit **TXC** možno použiť pri testovaní dokončenia všetkých prenosov na strane vysielateľa. Bit **RXC** možno použiť na otestovanie stavu: Na strane prijímateľa nie sú žiadne neprečítané dáta.

Poznámame, že **TXC** bit musí byť vynulovaný pred každým prenosom (predtým ako sa zapíše do **UDR**) ak je použitý k tomuto účelu.

USART vysielá a prijíma dátové rámce (data frames) ktoré sú tvorené z:

- 1 Start bit
- 5, 6, 7, 8 alebo 9 dátových bitov.
- Žiadnou, párnou, nepárnou... paritou. (**no, even, or odd parity bit**)
- 1 alebo dva stop bity.

Je jedno aký formát zvolíme, vysielateľ Tx a prijímač Rx musia byť konfigurované rovnako.

Najčastejší formát je: 1S, 8data, NP, 1P.

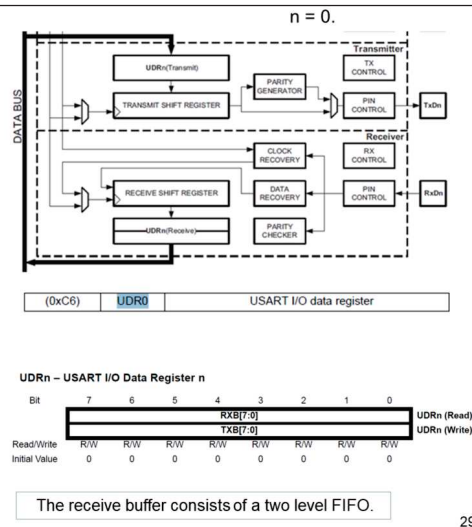
Komunikáciu zabezpečujeme cez TTL-USB prevodník signálov.

Najskôr musíme nastaviť baud rate. Default býva nastavená prenosová rýchlosť 9600 Bd.

### Vysielanie- USART0

```
void USART_Transmit( unsigned char data ){
/* Wait for empty transmit buffer */
while ( !( UCSR0A & (1<<UDRE0) ) );
/* Put data into buffer, sends the data */
UDR0 = data;
}
```

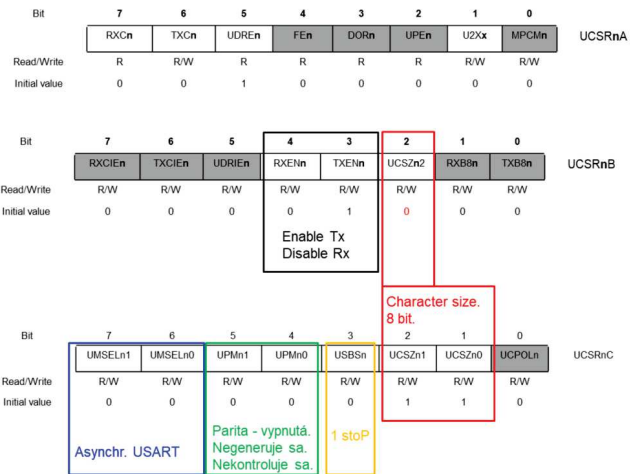
```
void zob_text_UART(char *s){
register unsigned char c;
while((c = *s++))USART_Transmit(c);
// retazec konci "nulou"
}
```



29

### Inicializácia - USARTn

AVR, ATMEGA 328P  
n = 0.



30

Vysielací register a prijímací register má jednu adresu. !?!

Vysielanie je jednoduché. „Stačí“ zapísať do vysielacieho registra ak sú preto splnené podmienky.

Prijímanie sa deje nasledovne. Po prijíme prvého platného stop bitu sa prepíše obsah posuvného registra do prijímacieho registra, do UDR0. Prijímací buffer je typu FIFO. Po prijíme jedného znaku by sme mali vykonať len jedno vyčítanie.

Vyšedené bity sú použité pri synchronnej, multiprocessorovej prevádzke, resp. vtedy ak použijeme prerušovací podsystem. Rovnako sa dnes nevenujeme tzv. chybovým príznakom.

Výpis dát na monitor PC predpokladá spustenie buď

- ARDUINO: -> Tools: -> Serial Monitor, 115200 b/s alebo

- PuTTY 115200 b/s .



Baud Rate (bps)	$f_{osc} = 16.0000 \text{ MHz}$			
	UBRRn	Error	UBRRn	Error
115.2k	8	-3.5%	16	2.1%
Max. (1)	1 Mbps		2 Mbps	

Na strane MMP nekomunikujeme z PC, ale s prevodníkom USB/USART.  
Softvér „USB“ spracuje nastavenú BR a nastaví parametre vysieláča prevodníka USB/USART pre  $f_{osc} = 16\text{MHz}$ .  
Soft má dve možnosti a vyberie tú menej zlú: U2X0 = 1 a UBRR0 = 16 s chybou +2,1%.

Na druhej strane, USART prijímač (programátor – JA sa práve naučil správne vypočítať číslo UBRR0 = 8 pre 115.2 kbs a  $f_{osc} = 16\text{MHz}$ ) a nastaví USB/USART  
Na horšiu možnosť: U2X0 = 0 a UBRR0 = 8 s chybou -3,5%.

Čo je celková chyba 5,6 %.

Pri testovaní komunikácie

- Chyba nastane občas ...
- Chyba nenastane ....
- Po čase niečo zmeníme a chyba opäť občas nastane ...

31

$T_{1b} = 8,68 \text{ us (115 200 bps)}$   
 $= 9,0 \text{ us (111 111,1 bps)}$   
 $= 8,5 \text{ us (117 647,1 bps)}$

UNO:  
BR = 117 647,1  
(U2X1 = 1 a UBRR0 = 16, chyba +2,1%)  
ZNAK = 0x55

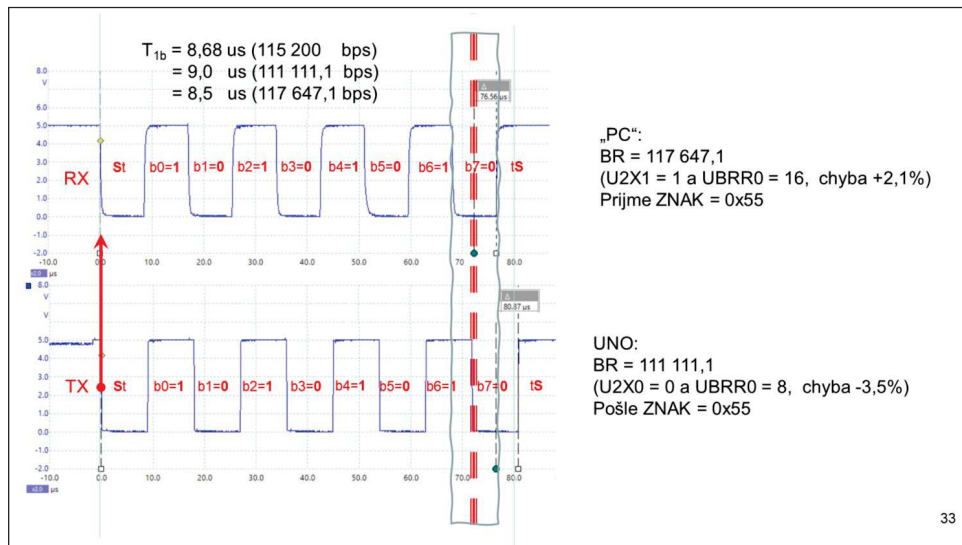
UNO:  
BR = 111 111,1  
(U2X0 = 0 a UBRR0 = 8, chyba -3,5%)  
ZNAK = 0x55

32

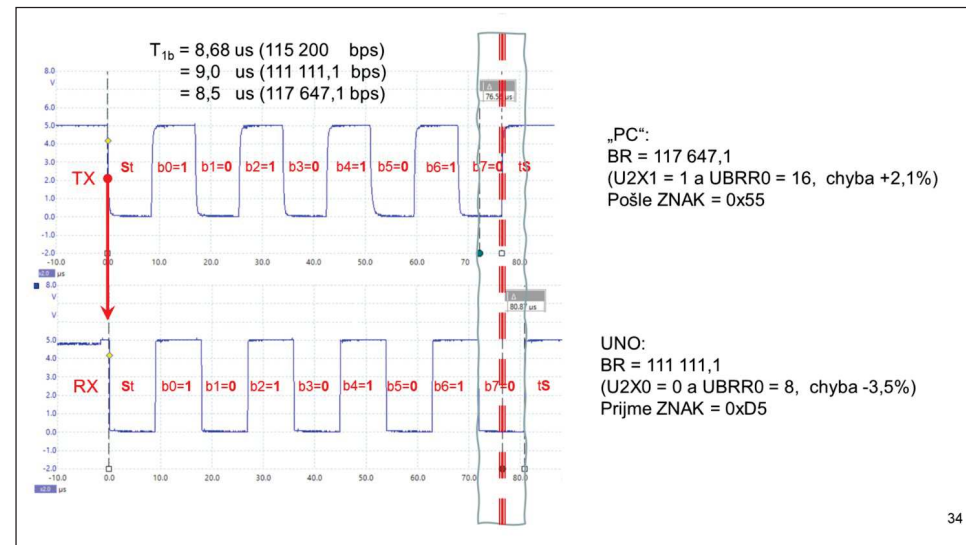
Doteraz všetky analýzy presnosti generovania BR boli založené na predpoklade, že jeden zo zdrojov je tzv. presný. Čo sa stane ak, sú oba nepresné? Vzorový príklad na sériovú komunikáciu predpokladá BR = 115200.

Podľa priloženej tabuľky môžeme nastaviť dve hodnoty registra UBRR0 pre U2X0=0, resp. 1.

Obe nastavenia USART-u komunikujú z „PC“ – ideálne nastavenie BR. Pri komunikácii N,115200,8,1 nenastane chyba komunikácie.



PC komunikuje cez USB port nastavený na BR = 117 647,1.



Správnu komunikáciu dostaneme ak aj na strane UNO nastavíme: BR tak, aby bola chyba +2,1%.