

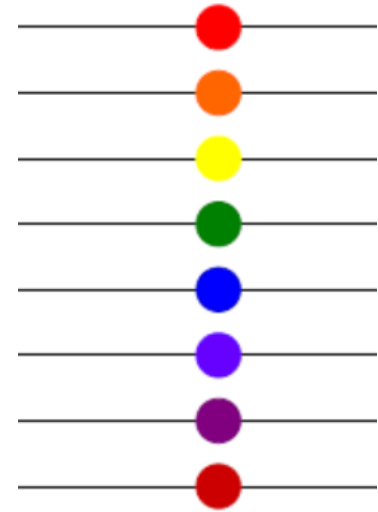
# Sériový přenos dat - USART

ATmega328P

# Sériový a paralelní přenos

## Paralelní přenos dat

- data jsou přenášena najednou po více bitech (obvykle podle šířky sběrnice)
- příslušný počet souběžných (paralelních) vodičů (a zemnicí vodič)
- vhodný na krátké vzdálenosti, přeslechy, časová neshoda jednotlivých signálů



## Sériový přenos dat

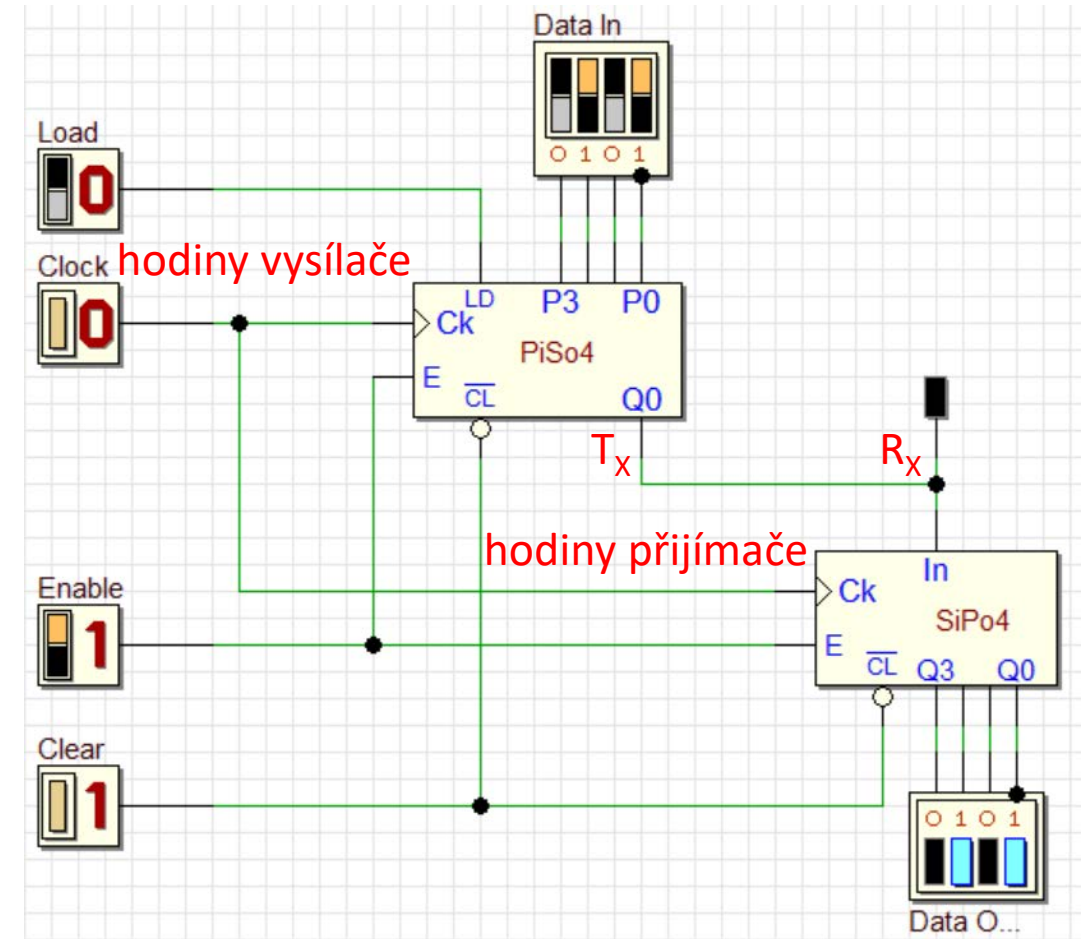
- data přenášena postupně bit po bitu po vysílacím (Tx) a přijímacím (Rx) vodiči, nebo symetrických párech
- vhodné na dlouhé vzdálenosti



# Synchronní přenos

## Synchronní přenos

- Má **jeden generátor** hodinových impulsů (na straně Data communication equipment – DCE) a **signál je přenášen** na druhou stranu, kde je **detekován** (na straně Data terminal equipment – DTE) .
- **Dávky datových bitů** následují **těsně po sobě** bez jakýchkoli časových odstupů a nejsou prokládány žádnými start- či stop-bity

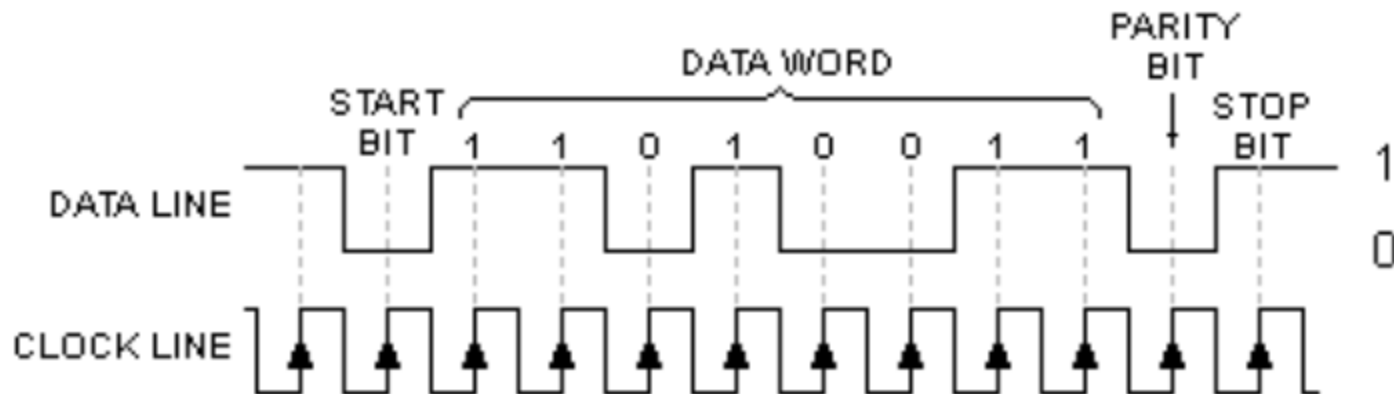


Synchronní spojení dvou posuvných registrů  $T_x$  a  $R_x$

# Asynchronní přenos

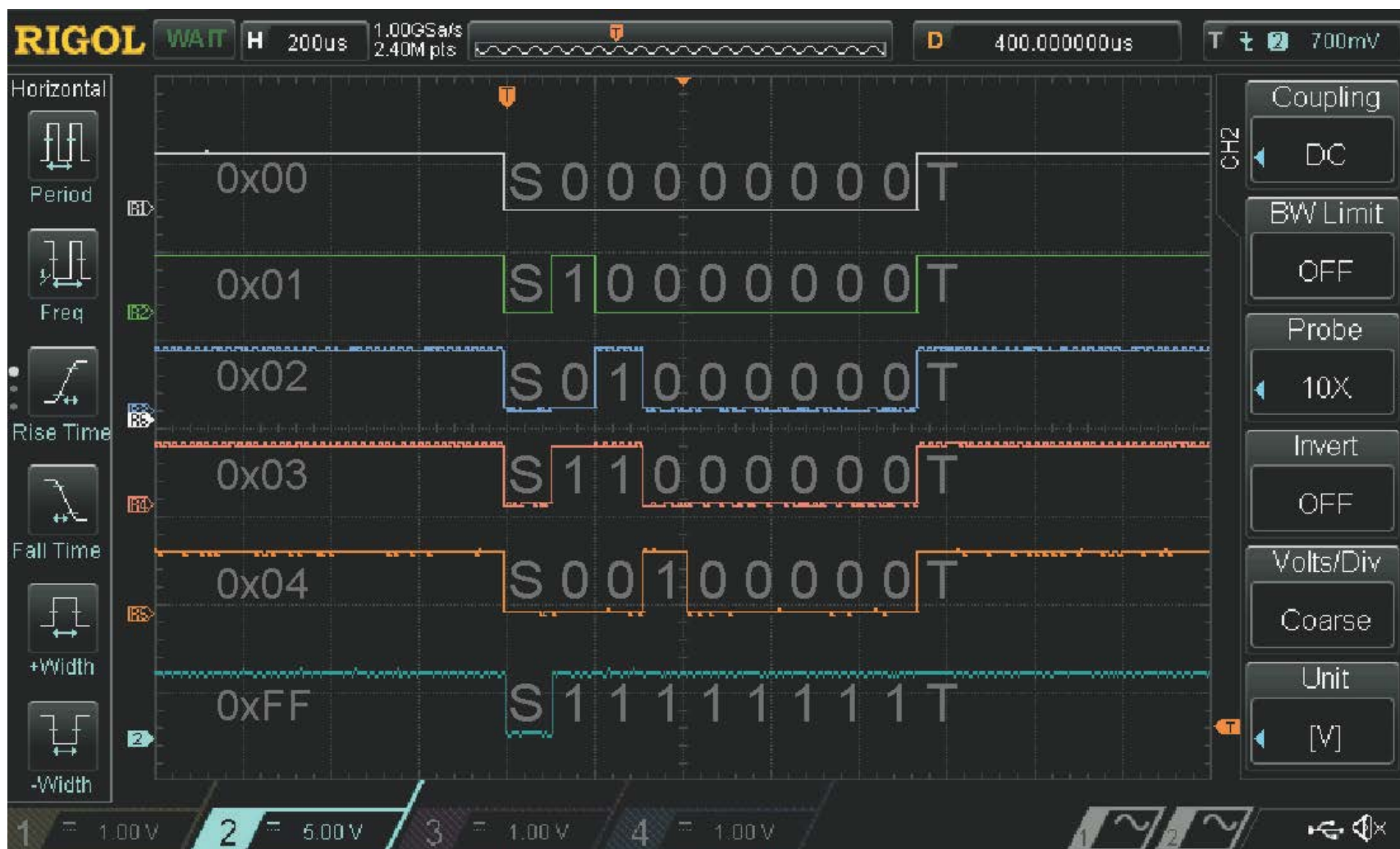
## Asynchronní přenos

- Má **na obou stranách vlastní generátor** hodinových impulsů nastavený na stejnou hodnotu rychlosti.
- **Hodinový signál se nepřenáší.**
- Jednotlivé datové bity jedné dávky jsou zapouzdřeny do **rámce** a jsou přenášeny **s libovolnými časovými odstupy** mezi sebou.



# Příklad sériové asynchronní komunikace

LSB se odesílá  
jako první po  
start bitu  
(data se čtou  
zprava)

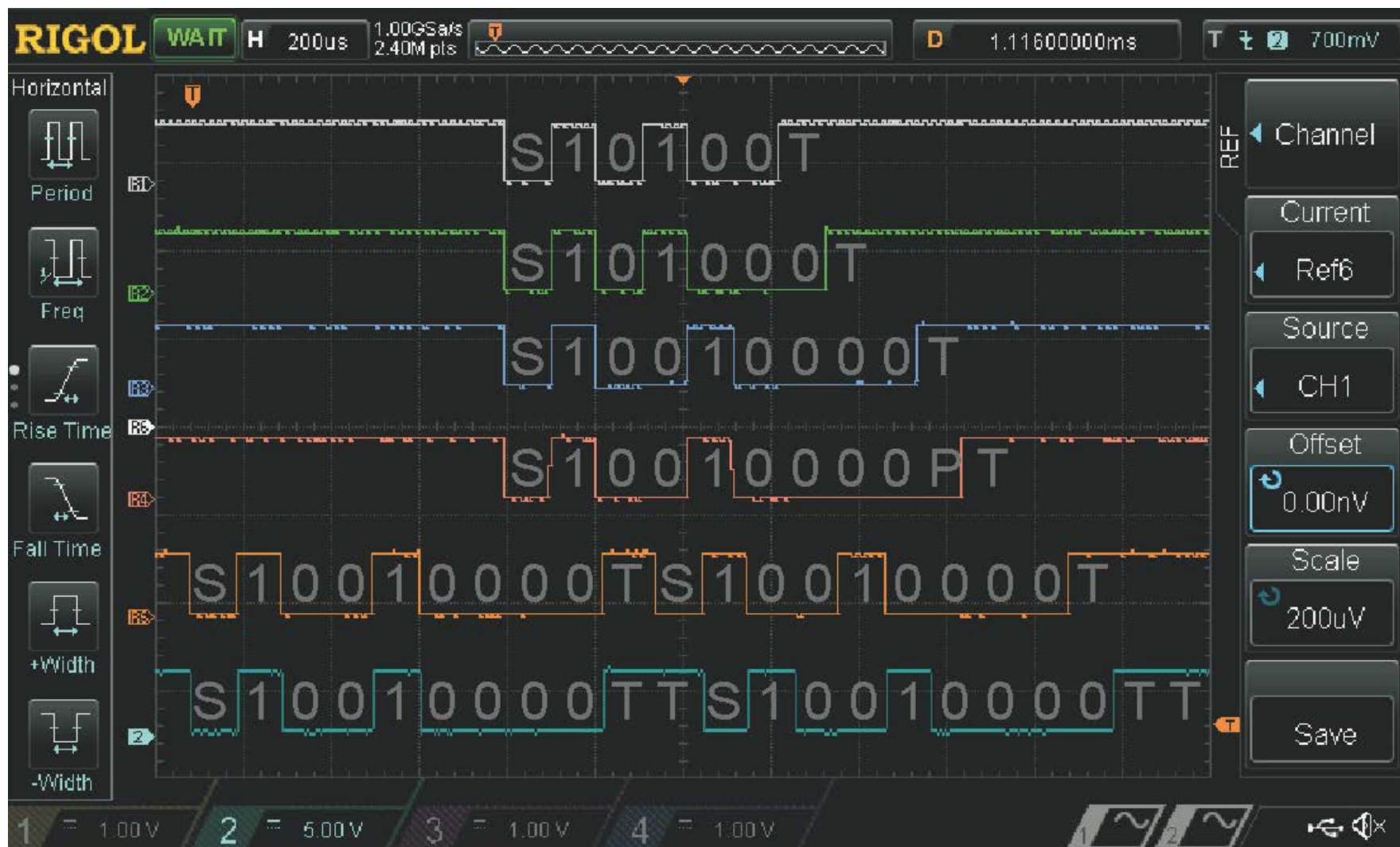


# Rychlost přenosu baudrate - bitrate

- Baud (Bd) - jednotka modulační rychlosti (také symbolová nebo znaková rychlost, anglicky baud rate)
- **Vztah mezi bitrate** (počtem přenesených bitů za sekundu) a modulační rychlostí **baudrate je dán způsobem kódování.**
- **Linkový kód** - způsob reprezentace digitálního signálu pro přenos médiem **v základním pásmu bez použití modulace**, kde platí  $1 \text{ baud} = 1 \text{ bit/s}$
- **USART** používá pro signály **TX a RX linkový kód** bez návratu k nule (Non Return to Zero – **NRZ**)  $LL = 0V$  a  $log1 = +U_{CC}$  kde **1 baud = 1 bit/s**

# Různá délka dat, parita, počet stopbitů

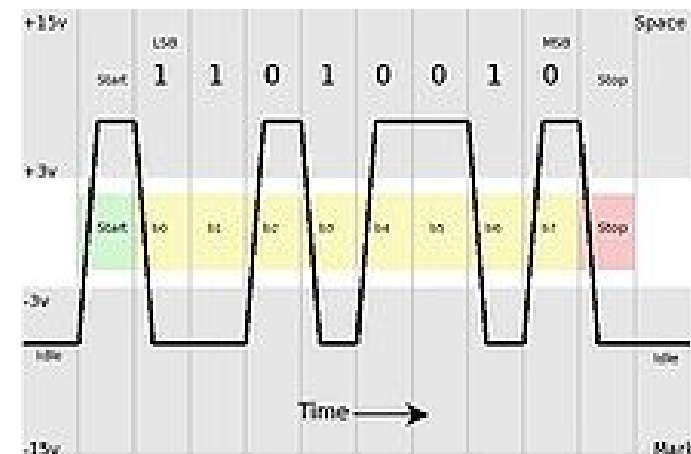
- bílá – 5bit, Žádná parita, 1 stop bit, data 0x05
- zelená – 6bit, Žádná parita, 1 stop bit, data 0x05
- modrá – 8bit, Žádná parita, 1 stop bit, data 0x09
- červená – 8bit, Sudá parita, 1 stop bit, data 0x09 (dvě zprávy)
- azurová – 8bit, Žádná parita, 2 stop bity, data 0x09 (dvě zprávy)
- Při datové rychlosti 9600Bd/s by měl jeden bit odpovídat  $1/9600 = 104\mu\text{s}$ .





# Logické úrovně signálů rozhraní UART

- sériový port (COM Port) – nutno změnit logické úrovně z mikrokontroléru (LL-0V, HL- $U_{CC}$  např. 5V) na logické úrovně RS232C (negativní logika, LL je +15V, HL je -15V). Např. obvod [MAX232](#)



- PC může komunikovat přes USB využitím virtuálního sériového portu (VCP, např. FTDI)

UART  $U_{CC} = 3,3V$  (LL = 0V, HL = 3,3V)

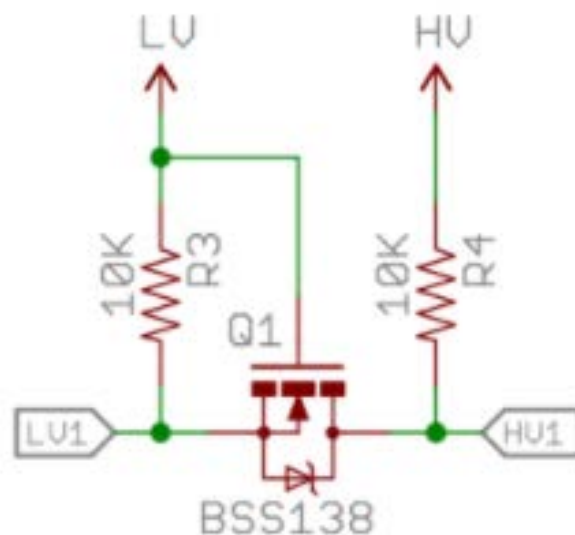
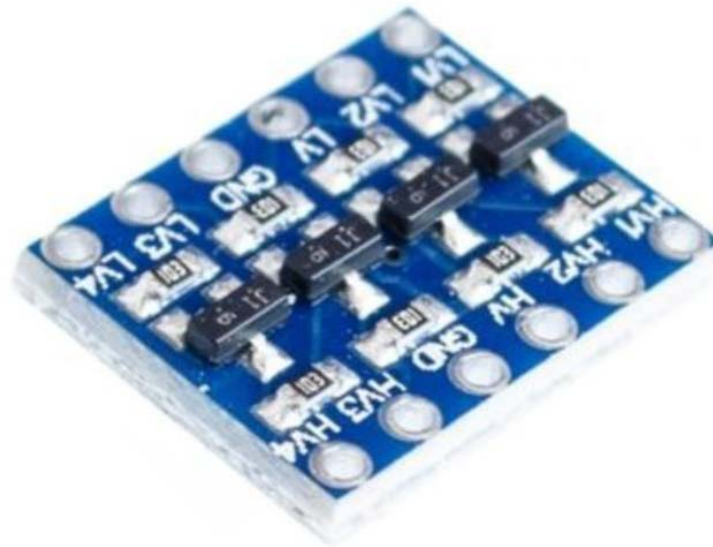
UART  $U_{CC} = 5V$  (LL = 0V, HL = 5V)



# Obousměrný převodník logických signálů

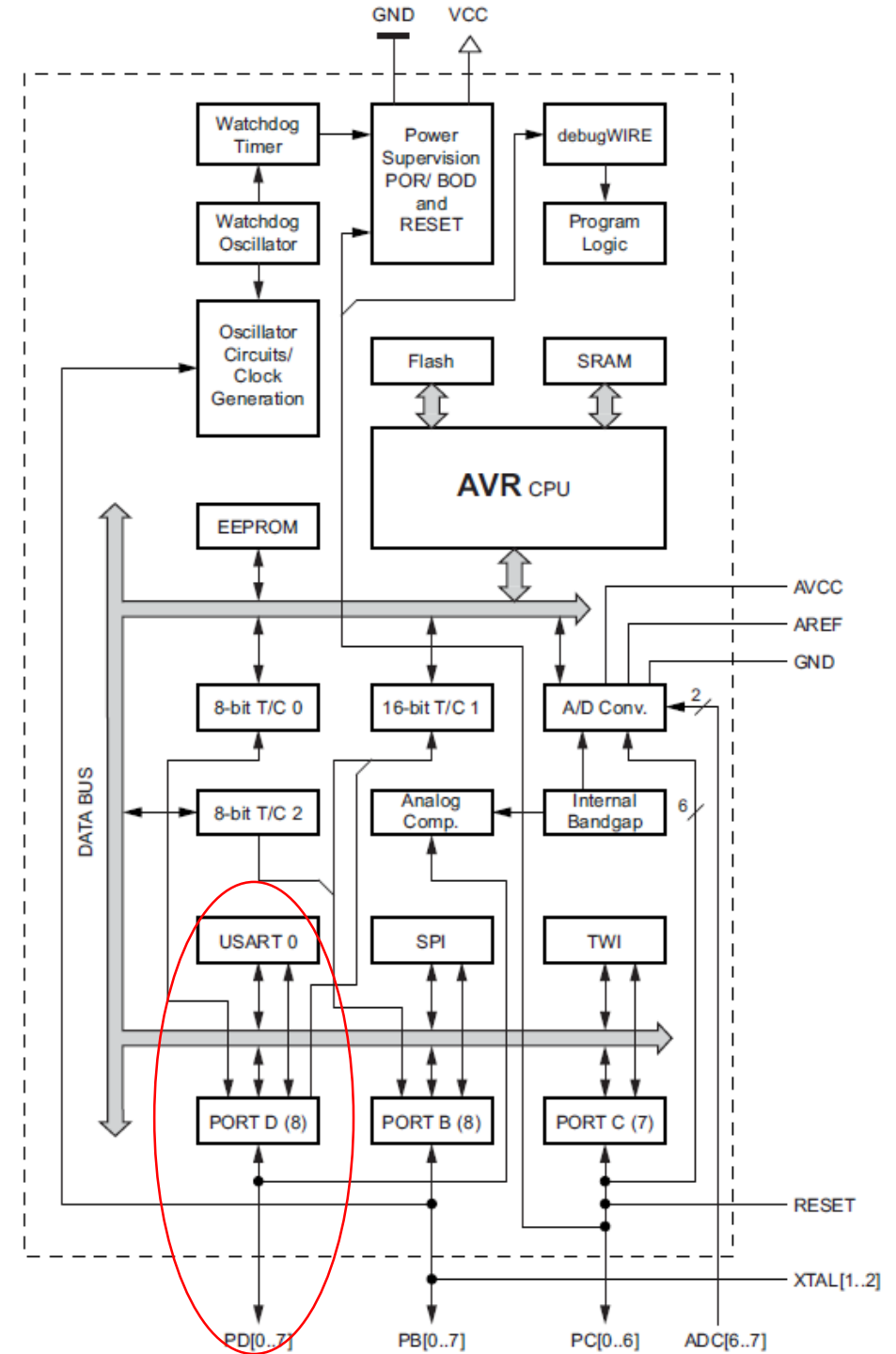
HV1	LV1
HV2	LV2
HV	LV
GND	GND
HV3	LV3
HV4	LV4

- např. [4 kanálové](#) propojení Arduino UNO s NodeMCU
- obousměrný převodník
- logické úrovně 5V a 3.3V



# USART ATmega328P

- **Plně duplexní** provoz (nezávislé sériové registry příjmu a vysílání)
- **Synchronní** provoz s taktováním **Master** nebo **Slave**
- **Asynchronní** provoz - sériové rámce – **start** bit, dále 5, až **8** nebo 9 **datových** bitů, (parita) a **1** nebo 2 **stop** bity
- Generování liché nebo sudé **parity** a kontrola parity podporovaná hardwarem
- Tři samostatná **přerušení** na **TX** kompletní, **TX** **datový** **registr** **prázdný** a **RX** kompletní
- Další výklad zaměřen na **USART 0** ( $n = 0$ )



# Nastavení režimu USART

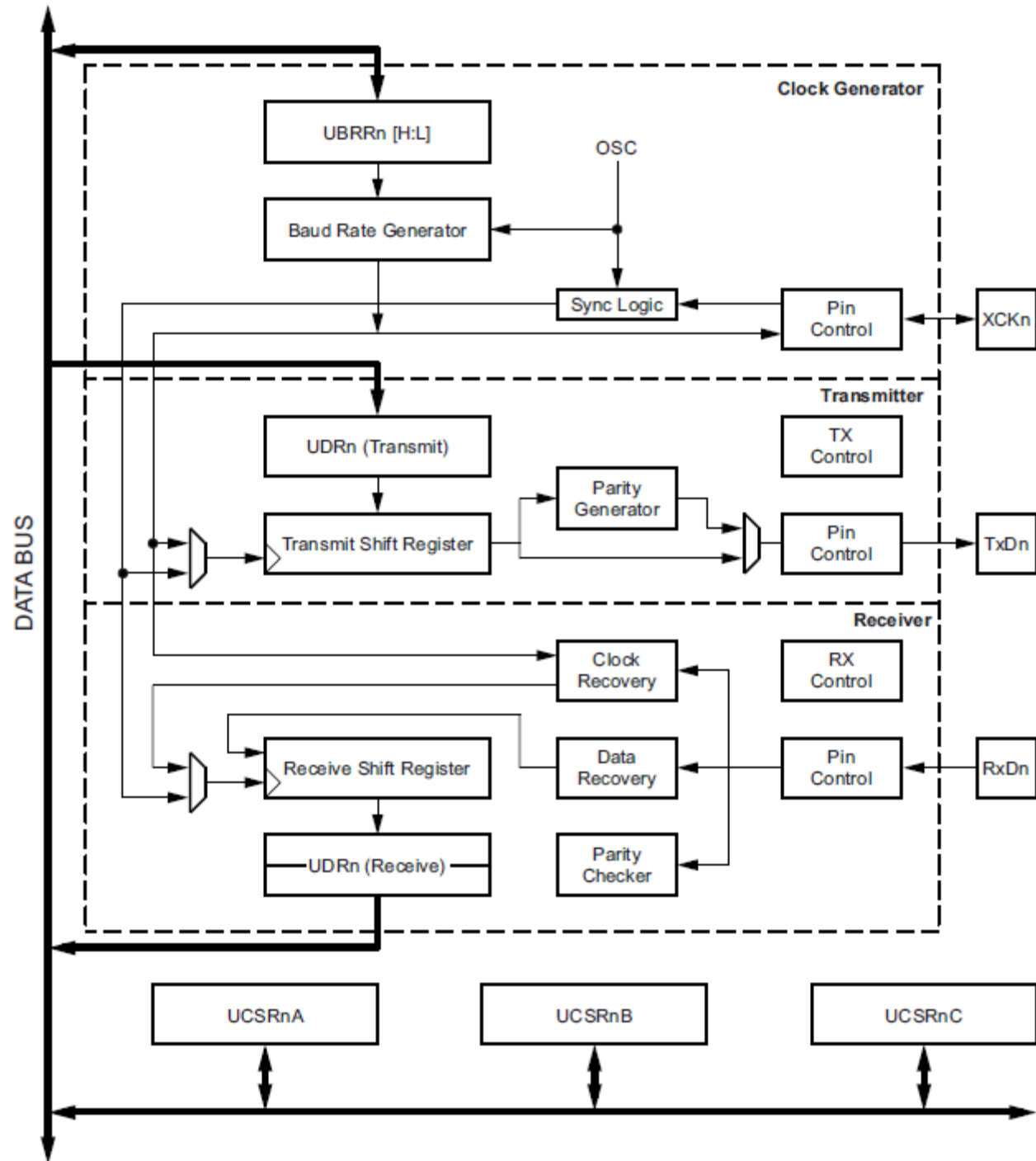
## Synchronní režim:

- USART může pracovat jako “**Master**” (generuje clock na pinu **XCK**) nebo jako “**Slave**” (přijímá clock z pinu **XCK**). Zapíná se nastavením bitu **UMSEL** v registru **UCSRC**
- **Rychlost přenosu** (baudrate) se nastavuje v registru **UBRR** (části *UBRRH* a *UBRRL*). Do nich je možné zapsat maximálně 12ti bitovou hodnotu (0 – 4095) dělicího poměru hodinového signálu. Dělení probíhá celočíselně (chyba by měla být max 1 %)
- V asynchronním režimu (bit **UMSEL** v registru **UCSRC** vynulován) je možné pomocí bitu **U2X** v registru **UCSRA** nastavit tzv. “**Double speed**”

# Asynchronní režim

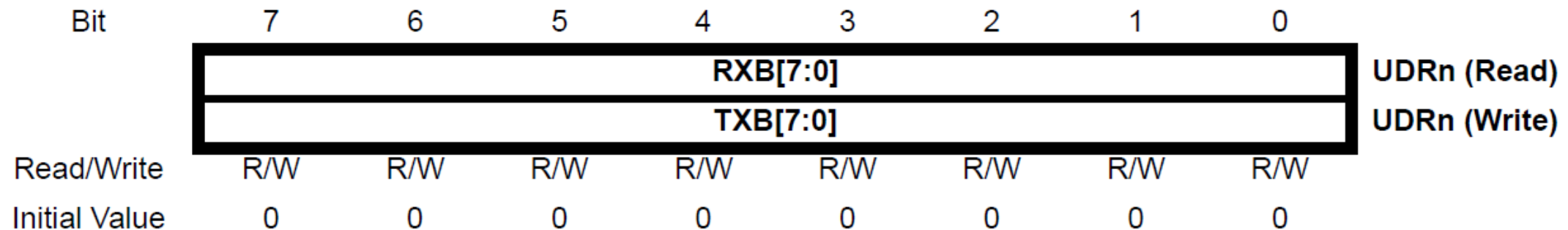
- Jednotku USART tvoří tři bloky: **generátor hodin, vysílač a přijímač**
- **Vysílač** tvoří jeden buffer, sériový posuvný registr, generátor parity a řídicí logiku pro různé formáty sériových rámců. Zapisovací buffer umožňuje kontinuální přenos rámců bez jakéhokoliv zpoždění mezi rámci.
- **Přijímač** obsahuje **navíc** jednotku pro **obnovu hodin** a dat, dále detektor parity, řídicí logiku, posuvný registr a dvouúrovňový přijímací kruhový buffer (*UDR*), který pracuje jako kruhový *FIFO* (*“First In – First Out”*).
- **Příznakové bity** chyb FE (**chyba rámce**), DOR (**ztráta znaku**) a devátý přijatý bit **RXB8** jsou ukládány s daty v přijímacím dvouúrovňovém bufferu UDR. Proto musí být vždy příznakové bity **přečteny dříve** než přenášená data.
- Přijímač podporuje ve stejném čase **stejně formáty rámců** jako vysílač, ale navíc může **detekovat chybu** rámce, ztráty znaku a **chybu parity**.

# Blokový diagram USART ATmega328P



# Registr UDR – datový buffer

- Je společný jak pro přijímač, tak pro vysílač.
- Abychom mohli do bufferu vysílače zapisovat, musí být bit UDRE v registru UCSRA nastaven (signalizuje, že vysílací buffer (UDR) je prázdný a připravený pro zápis).



LDS r16, UDR0 ; ulož přijatou hodnotu do r16

empty:

LDS r17, UCSR0A ; ulož UCSR0A do r17

SBRs r17, UDRE0 ; čekej na vyprázdnění vysílacího bufferu

rjmp empty ; repeat loop

STS UDR0, r16 ; vyšli hodnotu uloženou v r16

# Registr UCSRA – bity RXC, TXC, UDRE

Bit	7	6	5	4	3	2	1	0	
	<b>RXCn</b>	<b>TXCn</b>	<b>UDREn</b>	<b>FEn</b>	<b>DORn</b>	<b>UPEn</b>	<b>U2Xn</b>	<b>MPCMn</b>	UCSRnA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

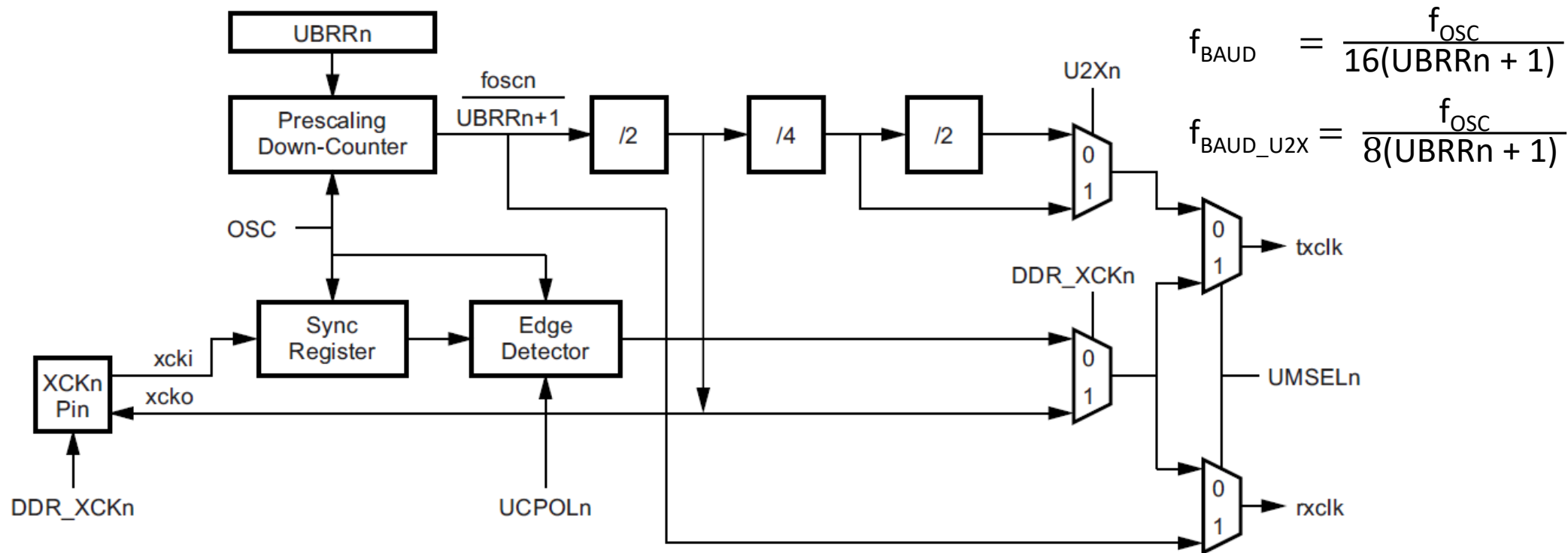
- Bity **RXC** a **TXC** slouží k informaci o **ukončení akce** příjmu, nebo vyslání znaku, po vyprázdnění bufferu dojde k automatickému vynulování bitu.
- **RXC, TXC** může generovat **přerušeni**
- **TXC** – nastaví se, pokud došlo k **odeslání všech dat** a již nejsou žádné data v bufferu vysílače.
- **UDRE** – nastaví se, když vysílací buffer (**UDR**) je **prázdný** a připravený pro zápis.
- Bit MPCM - nastavuje víceprocesorový režim, (výklad se tímto režimem nezabývá).



# Registr UCSRA – chybové stavy FE, DOR, PE

- Příznakové bity jsou **platné pouze do doby přečtení registru UDR**
- Při zápisu do registru UCSRA musí být vždy tyto bity nastaveny do stavu log. 0
- **Bit FE - indikace chyby rámce**, nastaví se, pokud nebyl indikován platný stop bit. K vynulování dojde po příjmu platného “stop bitu”.
- **Bit DOR - ztráta dat**, nastaví se, když je již přijímací buffer plný, tzn. obsahuje dva znaky, nový znak je v posuvném registru a je detekován nový “start bit”.
- **Bit UPE - chyba parity**, nastaví se, když nesouhlasí přijatá a vypočítaná parita.

# Časování přenosu - generátor hodin



txclk - hodiny vysílače

rxclk - hodiny přijímače

xcki - vstup z pinu XCK, synchronní slave

xcko - výstup hodin na pin XCK. synchronní master

fosc XTAL - systémové hodiny

U2X - dvojnásobná asynchronní přenosová rychlost

DDR\_XCK - data direction register XCK

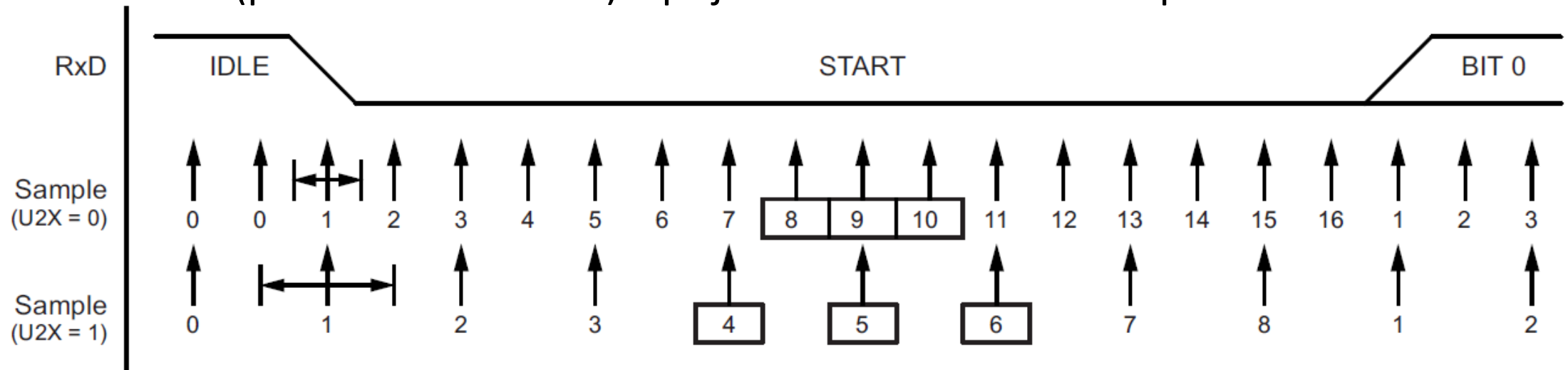
UMSEL - USART Mode Select (syn/asyn/SPI mód)

UCPOL - USART Clock Polarity (s UCPHA - časování SPI)

UBRR - USART baud rate register

# Logika obnovy hodinového signálu

- Po sestupné hraně v bodě 1 použije logika obnovy hodin vzorky 8, 9 a 10 pro normální režim a vzorky 4, 5 a 6 pro režim dvojnásobné rychlosti k rozhodnutí, zda je přijat platný počáteční bit.
- Pokud dva nebo více z těchto tří vzorků (většina) mají HL, je startovací bit odmítnut (považováno za šum) a přijímač začne hledat další přechod z HL na LL.



- Bit U2X registru UCSRA - nastavuje dvojnásobnou přenosovou rychlost asynchronního režimu. Nastavením změníme dělicí poměr pro přenosovou rychlost z 16 na 8.

# Výpočet přenosové rychlosti dané v UBRR

Operating Mode	Equation for Calculating Baud Rate <sup>(1)</sup>	Equation for Calculating UBRR Value
Asynchronous Normal Mode (U2X = 0)	$BAUD = \frac{f_{osc}}{16(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{16BAUD} - 1$
Asynchronous Double Speed Mode (U2X = 1)	$BAUD = \frac{f_{osc}}{8(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{8BAUD} - 1$
Synchronous Master Mode	$BAUD = \frac{f_{osc}}{2(UBRR + 1)}$	$UBRR = \frac{f_{osc}}{2BAUD} - 1$

# Nastavení předděliče pro používané hodnoty

Baud Rate (bps)	$f_{osc} = 16.0000\text{MHz}$			
	U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error
2400	416	-0.1%	832	0.0%
4800	207	0.2%	416	-0.1%
9600	103	0.2%	207	0.2%
14.4k	68	0.6%	138	-0.1%
19.2k	51	0.2%	103	0.2%
28.8k	34	-0.8%	68	0.6%
38.4k	25	0.2%	51	0.2%
57.6k	16	2.1%	34	-0.8%
76.8k	12	0.2%	25	0.2%
115.2k	8	-3.5%	16	2.1%
230.4k	3	8.5%	8	-3.5%
250k	3	0.0%	7	0.0%
0.5M	1	0.0%	3	0.0%
1M	0	0.0%	1	0.0%
Max. <sup>(1)</sup>	1Mbps		2Mbps	

Note: 1. UBRRn = 0, error = 0.0%

# Registr UCSRB – povolení RX, TX, přerušení

Bit	7	6	5	4	3	2	1	0	
	<b>RXCIE<sub>n</sub></b>	<b>TXCIE<sub>n</sub></b>	<b>UDRIE<sub>n</sub></b>	<b>RXEN<sub>n</sub></b>	<b>TXEN<sub>n</sub></b>	<b>UCSZ<sub>n2</sub></b>	<b>RXB8<sub>n</sub></b>	<b>TXB8<sub>n</sub></b>	UCSR <sub>n</sub> B
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

- **RXCIE** – přerušení **po příjmu dat** (pokud je nastaven RXC v registru UCSRA)
- **TXCIE** – přerušení **po dokončení vysílání dat** (pokud dojde k nastavení TXC v registru UCSRA)
- **UDRIE** – přerušení **po vyprázdnění registru UDR** (pokud dojde k nastavení UDRE v registru UCSRA).
- **RXEN** - nastavením dojde k **povolení přijímače** jednotky USART.
- **TXEN** - nastavením dojde k **povolení vysílače** jednotky USART.
- **UCSZ2** – určuje **délku datového znaku** (společně s **UCSZ1** a **UCSZ0** z registru UCSRC)
- **RXB8, TXB8** – devátý bit datového přenosu, nutno číst a zapsat před čtením a zápisem do UDRL. Ve výkladu neuvažujeme.

# Příklad povolení RX, TX, přerušení po příjmu

Bit	7	6	5	4	3	2	1	0	
	<b>RXCIE<sub>n</sub></b>	<b>TXCIE<sub>n</sub></b>	<b>UDRIE<sub>n</sub></b>	<b>RXEN<sub>n</sub></b>	<b>TXEN<sub>n</sub></b>	<b>UCSZ<sub>n2</sub></b>	<b>RXB8<sub>n</sub></b>	<b>TXB8<sub>n</sub></b>	<b>UCSR<sub>n</sub>B</b>
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Nastavení:

```
.org 0x0024          ; adresa vektoru přerušení 0x0024 (USART, Rx Complete, lze také zapsat .org URXCaddr)
rjmp int_rxc
```

```
LDI    r16, (1<<RXCIE0)|(1<<RXEN0)|(1<<TXEN0)    ; enable Rx interrupt, transmitter and receiver
STS    UCSR0B, r16
SEI                                ; enable global interrupt
```

Obsluha:

```
int_rxc:
    ; obsluha čtení Rx (případně opětovného poslání zpět Tx)
    ; popsáno u registru UDR
```

```
reti
```



# UCSRC – kontrolní a stavový registr C

Bit	7	6	5	4	3	2	1	0	
	<b>UMSELn1</b>	<b>UMSELn0</b>	<b>UPMn1</b>	<b>UPMn0</b>	<b>USBSn</b>	<b>UCSZn1</b>	<b>UCSZn0</b>	<b>UCPOLn</b>	<b>UCSRnC</b>
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

UCSZn2	UCSZn1	UCSZn0	Character Size
0	0	0	5-bit
0	0	1	6-bit
0	1	0	7-bit
0	1	1	8-bit
1	0	0	Reserved
1	0	1	Reserved
1	1	0	Reserved
1	1	1	9-bit

# Registr UCSRC - režim komunikace

- **UMSEL** - režim komunikace, nastavením je vybrán **synchronní** režim a vynulováním **asynchronní režim**. Pro zápis do registru UCSRC musí být bit URSEL nastaven.
- **UPM1 a UPM0 – parita**  
(even-sudá, odd-lichá)
$$P_{even} = d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 0$$
$$P_{odd} = d_{n-1} \oplus \dots \oplus d_3 \oplus d_2 \oplus d_1 \oplus d_0 \oplus 1$$
- **USBS** - počet “stop bitů”, nastavením použije dva “stop bity”, vynulováním pouze jeden “stop bit”.
- **UCPOL** – nastavení pouze v synchronním režimu - definuje vztah mezi datovým výstupem, vzorkováním vstupního signálu a hodinovým signálem na XCK. Při použití asynchronního režimu je tento bit vynulován.

# UCSRC – mód USARTu, parita

Bit	7	6	5	4	3	2	1	0	
	<b>UMSELn1</b>	<b>UMSELn0</b>	<b>UPMn1</b>	<b>UPMn0</b>	<b>USBSn</b>	<b>UCSZn1</b>	<b>UCSZn0</b>	<b>UCPOLn</b>	<b>UCSRnC</b>
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM) <sup>(1)</sup>

UPMn1	UPMn0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, even parity
1	1	Enabled, odd parity

# UCSRC – stop bit, polarita XCK

Bit	7	6	5	4	3	2	1	0	
	<b>UMSELn1</b>	<b>UMSELn0</b>	<b>UPMn1</b>	<b>UPMn0</b>	<b>USBSn</b>	<b>UCSZn1</b>	<b>UCSZn0</b>	<b>UCPOLn</b>	<b>UCSRnC</b>
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	1	1	0	

<b>USBSn</b>	<b>Stop Bit(s)</b>
0	1-bit
1	2-bit

<b>UCPOLn</b>	<b>Transmitted Data Changed (Output of TxDn Pin)</b>	<b>Received Data Sampled (Input on RxDn Pin)</b>
0	Rising XCKn edge	Falling XCKn edge
1	Falling XCKn edge	Rising XCKn edge

# Registry UBRRH a UBRRL - baudrate

Bit	15	14	13	12	11	10	9	8	
	–	–	–	–	UBRRn[11:8]				UBRRnH
	UBRRn[7:0]								UBRRnL
	7	6	5	4	3	2	1	0	
Read/Write	R	R	R	R	R/W	R/W	R/W	R/W	
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	
	0	0	0	0	0	0	0	0	

- USART Baud Rate Register - nastavení přenosové rychlosti (12 bitů – 8bitů UBRRL + 4 nižší UBRRH)

```
.equ ubp= (F_CPU/16/baud) – 1
LDI r16, LOW(ubp)
LDI r17, HIGH(ubp)
STS UBRR0L, r16
STS UBRR0H, r17
```

```
; USART baud předdělič (registr UBRR)
; nižší bity ubp do registru r16
; vyšší bity ubp do registru r16
; ulož baud předdělič do UBRR0
```