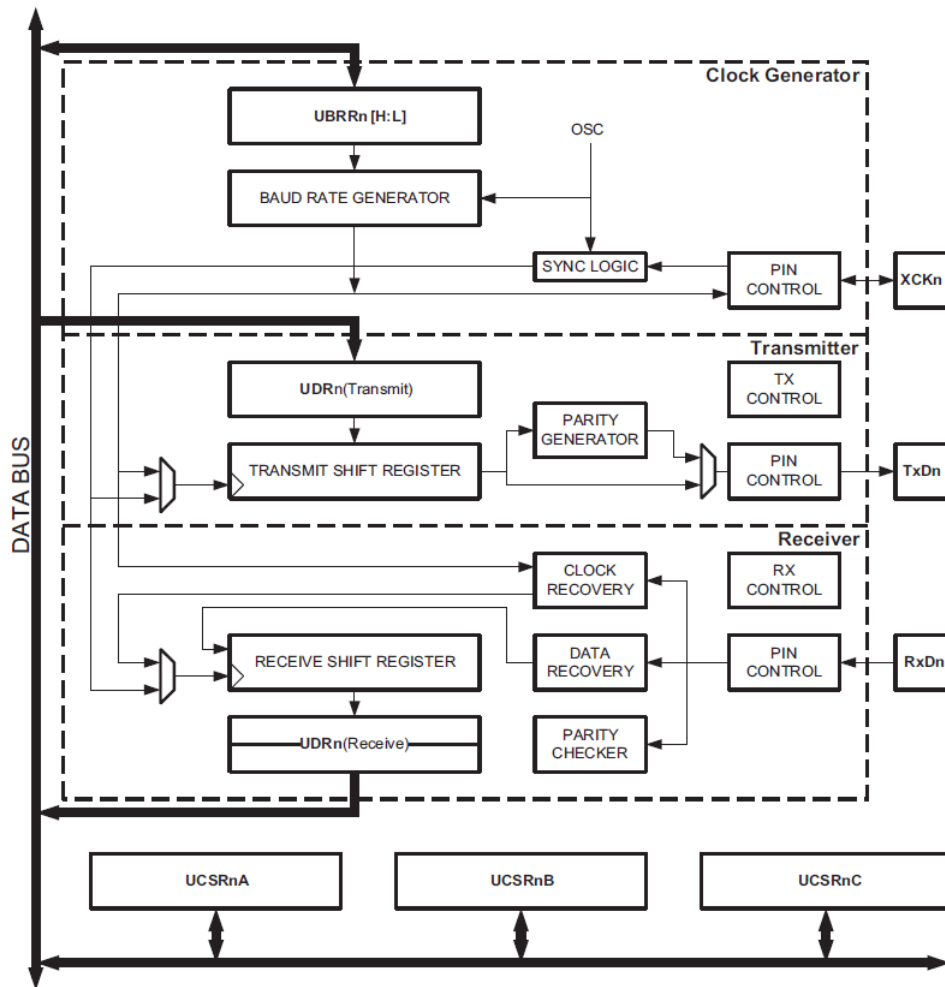


USART (Universal Synchronous and Asynchronous serial Receiver and Transmitter)

USART0 môže pracovať aj v režime „SPI MASTER“



Obr. Bloková schéma USART0

USART Je univerzálne sériové komunikačné zariadenie. Hlavné výhody:

- Duplexná prevádzka (nezávislé sériové prijímacie a vysielacie registre)
- Asynchrónna alebo synchronná prevádzka
- Podpora sériových rámcov s 5, 6, 7, 8 alebo 9 dátovými bitmi a 1 alebo 2 stop bitmi
- Tvorba párnej alebo nepárnej parity a kontrola parity pomocou hardwaru
- *Data OverRun Detection* – Kontrola pretečenia prijímacieho bafra
- *Framing Error Detection* – Kontrola chyby rámca
- Tri nezávislé prerušenia: *TX Complete*, *TX Data Register Empty* a *RX Complete*
- Multi-processorový mód komunikácie
- Asynchrónny komunikačný mód s dvojnásobnou rýchlosťou

USART je tvorený troma blokmi: generátor hodín (*Clock Generator*), vysielateľ (*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 bafru, 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 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.

Generátor prenosovej rýchlosti

Logika generovania taktovacieho signálu vytvára taktovacie signály pre vysielač aj prijímač. **USART** podporuje štyri módy generovania taktovacieho signálu:

Normal Asynchronous,

Double Speed Asynchronous,

(Master Synchronous a

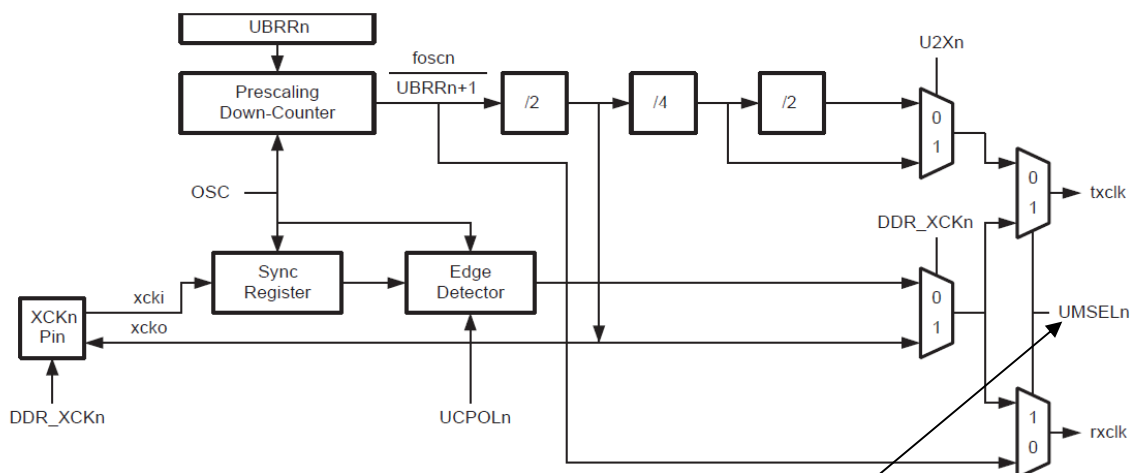
Slave Synchronous mode).

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

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

Bloková schéma obvodov generovania hodinových signálov je na ďalšom obrázku.

Clock Generation Logic, Block Diagram



Popis signálov:

txclk Transmitter clock (Vnútrotný signál).

rxclk Receiver base clock (Vnútrotný signál).

xcki Input from XCK pin (Vnútrotný signál). Použité len pri synchronnom **SLAVE** móde. **xcko** Clock output to XCK pin (Vnútrotný signál). Použité len pri synchronnom **MASTER** móde.

fosc XTAL pin frequency (Systémové hodiny).

Generátor vnútorných hodinových signálov je použitý aj pre asynchrónnu prevádzku aj pre synchronný 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 **UBRRn**. Taktovací signál sa generuje pokaždé, keď počítadlo dosiahne nulu. Tieto hodiny sú výstupom generátora taktovacej frekvencie prenosovej rýchlosti ($= f_{osc}/(UBRRn+1)$). Vysielač 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.

Pozor niekedy to vyzerá zmetočne: Pozri obr. vyššie a riadiaci register nižšie.

7	6	5	4	3	2	1	0	
UMSELn1	UMSELn0	UPMn1	UPMn0	USBSn	UCSZn1	UCSZn0	UCPOLn	UCSRnC

V tabuľke sú zapísané vzťahy výpočtu prenosovej rýchlosti (v bitoch za sekundu [bps]) a výpočtu hodnoty **UBRR** pre každý mód činnosti využívajúci vlastné taktovanie.

Equations for Calculating Baud Rate Register Setting

Operating Mode	Equation for Calculating Baud Rate ⁽¹⁾	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	$BAUD = \frac{f_{OSC}}{2(UBRRn + 1)}$	$UBRRn = \frac{f_{OSC}}{2BAUD} - 1$

Poznámka: 1.

BAUD - Baud rate (v bitoch za sekundu, [bps])

f_{OSC} - frekvencia systémového oscilátora

UBRR - Obsahy registrov UBRRH a UBRRL, (0 - 4095)

Niekoľko príkladov výpočtu hodnôt UBRR vid'. (Tabuľka na str. 196).

Double Speed Operation (U2Xn)

Prenosovú rýchlosť (asynchrónny mód) možno zdvojnásobiť nastavením bitu **U2Xn** v **UCSRnA**. Pre synchronnú prevádzku nastavte **U2Xn=0**.

!!! Ak nastavíme **U2Xn** pri asynchrónnej prevádzke, prenosová rýchlosť musí byť presnejšia. Väčší problém pri obnove dát a odchytení štart bitu. Pre vysielač nie sú žiadne obmedzenia z tohto titulu.

External Clock Používajú sa len pri synchronnom slave móde. Neuvažujeme.

Synchronous Clock Operation Neuvažujeme.

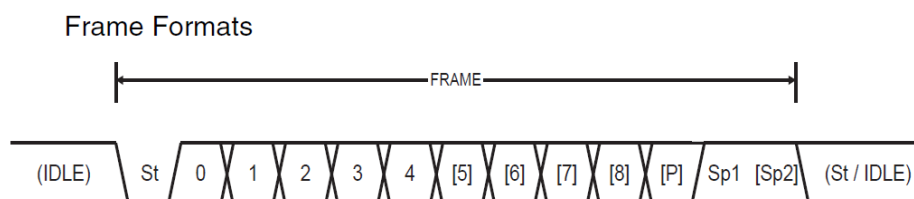
Formát prenosových rámcov

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.)

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 kľudového stavu – vysoká úroveň – log. 1.

Kombinácie formátov rámcov sú zrejme 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 páru alebo nepáru 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 – vkľude. 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 ktoréhokolvek bitu spôsobí zrušenie 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

Paritný bit je počítaný pomocou operácie exclusive-or zo všetkých dátových bitov. Ak použijeme nepárnu paritu, výsledok je invertovaný. Vzťah paritného bitu a dátových bitov je nasledovný:

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

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 USARTu

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 prenosi. Bit **TXC** možno použiť pri testovaní dokončenia všetkých prenosov na strane vysielača. Bit **RXC** možno použiť na otestovanie stavu: Na strane prijímača nie sú žiadne neprečítané dáta. Poznamenajme, že **TXC** bit musí byť vynulovaný pred každým prenosom (predtým ako sa zapíše do **UDR**) ak je použitý k tomuto účelu. **Ak by do pôvodného textu napísali: Zápis do UDR sa vykoná po predchádzajúcom vynulovaní bitu TXC mohli vynechať dosť veľkú časť popisu USARTu. Ak nieje bit TXC vynulovaný, zápis do UDR sa nevykoná.**

Nasledujúci program v C-ku ukazuje inicializáciu asynchrónneho prenosu bez prerušenia – „polling“. Prenosová rýchlosť je daná ako parameter funkcie.

Pri prechode k ATMEGA 8: POZOR!!!!

Dva **UBRRH** a **UCSRC** majú tú istú I/O adresu.

Ak funkcia zapisuje do registra **UCSRC**, bit **URSEL** (MSB) musí byť nastavený. Dôvod: Pri zápise ich dokážeme odlíšiť pomocou najvyššieho bitu. Ak **URSEL** = 1 zapisujeme do **UCSRC** a ak **URSEL** = 0 zapisujeme do registra **UBRRH**.

C Code Example⁽¹⁾

```
#define FOSC 1843200 // Clock Speed
#define BAUD 9600
#define MYUBRR FOSC/16/BAUD-1
void main( void )
{
  ...
  USART_Init(MYUBRR)
  ...
}
void USART_Init( unsigned int ubrr)
{
  /*Set baud rate */
  UBRR0H = (unsigned char)(ubrr>>8);
  UBRR0L = (unsigned char)ubrr;
  /* Enable receiver and transmitter */
  UCSR0B = (1<<RXEN0)|(1<<TXEN0);
  /* Set frame format: 8data, 2stop bit */
  UCSR0C = (1<<USBS0)|(3<<UCSZ00);
}
```

Poznámka: 1. Príklad predpokladá, že potrebné hlavičkové súbory sú pripojené.

(0xC5)	UBRR0H						USART Baud Rate Register High			
(0xC4)	UBRR0L						USART Baud Rate Register Low			
(0xC3)	Reserved	–	–	–	–	–	–	–	–	
(0xC2)	UCSR0C	UMSEL01	UMSEL00	UPM01	UPM00	USBS0	UCSZ01 / UDOR00	UCSZ00 / UCPHA0	UCPOL0	
(0xC1)	UCSR0B	RXCIE0	TXCIE0	UDRIE0	RXEN0	TXEN0	UCSZ02	RXB80	TXB80	

Vysielanie Dát – USART vysielateľ

Vysielateľ USART-u zapneme nastavením bitu *Transmit Enable (TXEN)* v registri **UCSRnB**. V okamžiku zapnutia vysielateľa sa normálna prevádzka pinu **TxD** preruší a pin je ovládaný USART-om. Pri synchronnej prevádzke je USART-om riadený aj pin **XCK**.

Poslanie rámca, s 5 až 8 dátových bitov

Prenos dát sa zahajuje zápisom vysielaných dát do vysielacieho bafra. CPU naplní vysielací bafer zápisom do registra **UDR** v priestore I/O. Dáta z vysielacieho bafra budú presunuté do posuvného vysielacieho registra keď bude tento pripravený odvysielať ďalší rámec. Posuvný register sa naplní novými dátami ak je v *idle* stave alebo po odvysielaní posledného stop bitu predchádzajúceho rámca. Po naplnení posuvného registra sa odvysiela celý rámec nastavenou prenosovou rýchlosťou.

Nasledujúci program ukazuje jednoduchý prenos pomocou USART využívajúci techniku: testovanie stavu - polling. Testujeme príznak: *Data Register Empty (UDREN)*. Keď použijeme prenosový rámec s menším počtom bitov ako osem, najvýznamnejšie bity zapísané do registra **UDRn** sa ignorujú. USART treba inicializovať pred vysielaním.

```

C Code Example(1)

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

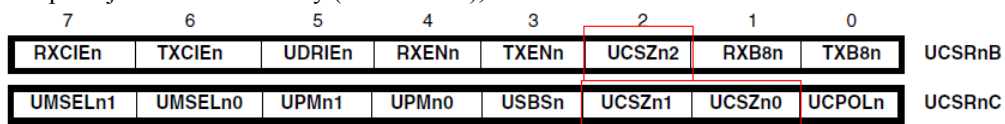
```

Poznámka: 1. Príklad predpokladá, že potrebné hlavičkové súbory sú pripojené.

Program čaká na ukončenie predchádzajúceho prenosu testovaním príznaku **UDRE**. Potom zapíše ďalšie dáta do vysielacieho bafra. Ak použijeme prerušenie od príznaku *Data Register Empty* prerušovací podprogram zapíše do registra ďalšie vysielané dáta.

Poslanie rámca, 9 dátových bitov

Ak použijeme 9-bitové znaky (**UCSZn = 7**),



deviaty bit musí byť zapísaný do bitu **TXB8_n** v registri **UCSRnB** predtým ako zapíšeme spodný byte do **UDRn**. Nasleduje program pre odovšielanie 9-bitových znakov.

```

C Code Example(1)(2)

void USART_Transmit( unsigned int data )
{
    /* Wait for empty transmit buffer */
    while ( !( UCSRnA & (1<<UDREN)) )
        ;
    /* Copy 9th bit to TXB8 */
    UCSRnB &= ~(1<<TXB8);
    if ( data & 0x0100 )
        UCSRnB |= (1<<TXB8);
    /* Put data into buffer, sends the data */
    UDRn = data;
}

```

Deviaty dátový bit možno použiť, napr. na označenie rámca adresy pri multiprocesorovej komunikácii.

Stavové bity vysielacza USARTu, prerušenia

Vysieláč USART-u obsahuje dva príznakové bity: *USART Data Register Empty* (**UDREN**) a *Transmit Complete* (**TXCN**). Oba bity možno použiť pri generovaní prerušenia.

Príznak **UDREN** indikuje, či je vysielací bafer pripravený prijať ďalšie dáta. Tento bit sa nastaví ak je prijímací bafer prázdny. A vynuluje sa ak obsahuje dáta, ktoré sa ešte nepresunuli do posuvného registra. Z dôvodu budúcej kompatibility vždy zapisujte do tohto bitu nulu ak zapisujete do **UCSRA** registra.

Ak bit *Data Register empty Interrupt Enable* (**UDRIEN**) v registri **UCSRnB** nastavíme na jednotku, prerušenie typu *USART Data Register Empty* sa bude vykonávať tak dlho pokiaľ bude **UDREN** nastavené (Predpokladáme nastavenie bitu globálne prerušenie). **UDREN** sa vynuluje zápisom do **UDRn**. Ak použijeme prerušením riadené vysielanie, podprogram obsluhy prerušenia „*Data Register Empty Interrupt*“ musíme v prerušení buď zapísať do **UDR** nové dáta, aby sme vynulovali **UDRE** alebo musíme vypnúť prerušenie od „*Data Register empty*“. Ináč sa okamžite vrátíme do prerušenia po jeho ukončení.

Príznak *Transmit Complete* (**TXCN**) sa nastaví ak sa odvysielala celý rámec cez posuvný register a vo vysielacom bafre nie sú žiadne ďalšie dáta. Príznak **TXCN** sa automaticky vynuluje ak sa objaví prerušenie typu: „*transmit complete interrupt*“, alebo ho možno vynulovať zápisom jednotky na pozíciu tohoto bitu. Príznak **TXCN** sa s výhodou používa pri *half-duplex* komunikácii (napr. Pri štandarde RS485), kde odvysielanie musí vyvolať prijímací mód a uvoľniť komunikačnú zbernicu okamžite po ukončení vysielania.

Ak je nastavený bit *Transmit Complete Interrupt Enable* (**TXCIEN**) v registri **UCSRnB**, **USART** prerušenie *Transmit Complete Interrupt* sa vykoná ak sa príznak **TXCN** nastaví (Predpokladáme nastavenie bitu globálne prerušenie). Ak použijeme mód *transmit complete interrupt*, obsluha prerušenia nemusí nulovať príznak **TXC**, toto sa udeje automaticky pri vykonaní prerušenia.

Paritný generátor

Generátor parity vypočítava paritný bit pre sériový rámec. Ak je paritný bit povolený (**UPMn1** = 1), vysielacia logika vloží paritný bit medzi posledný dátový bit a prvý **stoP** bit.

Vypnutie vysielacza

Vypnutie vysielacza (nastavenie **TXEN** na nulu) zaúčinkuje až vykonávaný a požadovaný prenos sa ukončí, t.j., ak posuvný vysielací register a vysielací bafer register nebudú obsahovať dáta určené k odvysielaniu. Ak vysieláč vypneme pin **TxD** prestane byť riadený **USART**-om.

USART – Prijímač dát

Prijímač USART-u sa zapne nastavením bitu *Receive Enable* (**RXENn**) v registri **UCSRnB**. Keď je prijímač USART-u zapnutý riadenie pinu **RxD** preberie **USART**. Formát prijímaného rámca a prenosová rýchlosť musí byť nastavená pred prvým príjmom. Pri synchrónnej prevádzke pin **XCK** prijíma taktovacie signály.

Príjem rámca, 5 až 8 dátových bitov

Prijímač zahájí príjem po detekovaní platného štart bitu. Každý nasledujúci bit za štart bitom bude vzorkovaný pomocou prenosovej rýchlosti alebo hodín prichádzajúcich cez pin **XCK** a posunutý do posuvného registra. **Vzorkovanie a posúvanie sa ukončí prijatím prvého stoP bitu rámca**. Druhý **stoP** bit je prijímačom ignorovaný. V okamžiku prijatia platného **stoP** bitu sa obsah posuvného registra prekopíruje do prijímacieho bafra. Prijímací bafer možno prečítať čítaním registra **UDRn** v I/O priestore.

Nasledujúci program ukazuje jednoduchý príjem pomocou **USART** využívajúci techniku: testovanie stavu - *polling*. Testujeme príznak: *Receive Complete* (**RXCn**). Keď použijeme prenosový rámec s menším počtom bitov ako osem, najvýznamnejšie bity čítané z registra **UDRn** sa maskujú na nulu. **USART** treba inicializovať pred vysielaním.

C Code Example⁽¹⁾

```
unsigned char USART_Receive( void )
{
    /* Wait for data to be received */
    while ( !(UCSRnA & (1<<RXcN)) )
        ;
    /* Get and return received data from buffer */
    return UDRn;
}
```

Poznámka: 1. Príklad predpokladá, že potrebné hlavičkové súbory sú pripojené.

Príjem rámcu, 9 dátových bitov

Pri prenose 9-bitových znakov ($UCSZn = 7$), deviaty bit musí byť prečítaný z bitu **RXB8n** v registri **UCSRnB** pred prečítaním spodných bitov **UDRn**. Toto pravidlo platí aj pre príznaky **FEn**, **DORn** a **PEn**. Najskôr vyčítame stav z **UCSRnA**, a potom dáta z **UDRn**. Čítanie **UDRn** z I/O pozície zmení stav prijímacieho FIFO bafru a odpovedajúco bity **TXB8n**, **FEn**, **DORn** a **PEn**, ktoré sú tiež uložené vo FIFO.

Nasledujúci program ukazuje príjem 9-bitových dát. Program spracováva aj 9-bitové dáta aj stavové bity. Program najskôr vyčíta všetky I/O registre a potom ich spracuje. Takýto postup uvoľní prijímací bafer pre ďalšie dáta.

C Code Example⁽¹⁾

```
unsigned int USART_Receive( void )
{
    unsigned char status, resh, resl;
    /* Wait for data to be received */
    while ( !(UCSRnA & (1<<RXcN)) )
        ;
    /* Get status and 9th bit, then data */
    /* from buffer */
    status = UCSRnA;
    resh = UCSRnB;
    resl = UDRn;
    /* If error, return -1 */
    if ( status & (1<<FEn) | (1<<DORn) | (1<<UPEn) )
        return -1;
    /* Filter the 9th bit, then return */
    resh = (resh >> 1) & 0x01;
    return ((resh << 8) | resl);
}
```

1. See "About Code Examples" on page 7.

For I/O Registers located in extended I/O map, "IN", "OUT", "SBIS", "SBIC", "CBI", and "SBI" instructions must be replaced with instructions that allow access to extended I/O. Typically "LDS" and "STS" combined with "SBRS", "SBRC", "SBR", and "CBR".

Poznámka: 1. Príklad predpokladá, že potrebné hlavičkové súbory sú pripojené.

Stavové bit prijímača USARTu, prerušenia

Prijímač USART-u má príznak, ktorý indikuje stav prijímača. Príznak *Receive Complete* (**RXCn**) indikuje či sú v prijímačom bafre neprečítané dáta. Tento príznak je jednotkový, ak sú v prijímačom bafre neprečítané dáta a nulový ak je prijímací bafer prázdny. Ak je prijímač vypnutý (**RXENn** = 0), prijímací bafer sa vyprázdni a následne sa vynuluje **RXCn**. Ak je prerušenie povolené, tj. ak je *Receive Complete Interrupt* (**RXCIE**n) v registri **UCSRnB** nastavené, prerušenie od *USART Receive Complete Interrupt* sa bude vykonávať pokiaľ bude **RXCn** nastavené (predpokladáme, že **GI** je povolené). Ak použijeme režim príjem dát riadený prerušením, prerušovacia rutina musí vyčítať register **UDRn**, čím vynuluje **RXCn** príznak, ináč sa objaví ďalšie prerušenie po ukončení obsluhy prerušenia.

Chybové príznaky prijímača USARTu

Prijímač USART-u má tri chybové príznaky: *Frame Error* (**FEn**), *Data OverRun* (**DORn**) a *Parity Error* (**PE**n). Všetky sú prístupné čítaním **UCSRnA**. **Spoločné pre Error Flags – tieto tri bity, je to, že sú umiestnené v prijímačom bafre spolu s odpovedajúcim rámcom.** Z tohoto dôvodu chybové príznaky musia byť čítané – (**UCSRnA**) pred čítaním (**UDRn**). Dôvod: čítanie I/O registra **UDRn** zmení smerník čítania bafra – „ďalšie čítanie **UDRn** bude z iného miesta“. Pri zápise do **UCSRnA** treba do týchto bitov zapisovať nuly. Žiaden chybový príznak nemôže generovať prerušenie.

Príznak *Frame Error* (**FEn**) indikuje stav prvého stop bitu posledne čítaného bitu. Ak bol stop bit prijatý korektne, tj. mal hodnotu jedna, príznak **FEn** bude mať hodnotu 0, a naopak. Pri zápise do **UCSRnA** vždy nastavte tento bit na nulu.

Príznak *Data OverRun* (**DORn**) indikuje stratu dát zapríčinenú preplnením prijímacieho bafra. Príznak *Data OverRun* sa nastaví, ak je prijímací bafer plný (dva znaky), v prijímačom posuvnom registri je znak a objaví sa ďalší Start bit. Z dôvodu budúcej kompatibility nastavte tento bit na nulu pri zápise do **UCSRnA**. Príznak **DORn** sa nuluje ak sa úspešne presunie obsah posuvného registra do prijímacieho bafra.

Príznak *Parity Error* (**PE**n) indikuje chybu parity prijatého rámca. Ak nie je kontrola parity nastavená, tento bit čítame ako nulu. Z dôvodu budúcej kompatibility nastavte tento bit na nulu pri zápise do **UCSRnA**.

Kontrola parity

Kontrola parity je aktívna, ak je horný bit *USART Parity mode* (**UPMn1**) nastavený na jedna. Typ kontrolovanej parity (párna alebo nepárna) sa nastavuje bitom **UPMn0**. Ak je kontrola nastavená, obvod kontroly priebežne vypočítava paritu na strane prijímača a po príchode paritného bitu tieto porovná. Výsledok porovnania sa uloží do bitu *Parity Error* (**PE**n) spolu s prijatými datami a stop bitmi. Bit **PE**n možno prečítať ak je **UPMn1** = 1. Tento bit je platný do okamžiku prečítania prijímacieho bafra (**UDRn**).

Vypnutie prijímača USARTu

Na rozdiel od vysielacza, vypnutie prijímača je okamžité. Práve prijímané dáta teda budú stratené. V okamžiku vypnutia (tj. **RXENn** = 0) **USART** prestane riadiť pin **RxDn** a tento prejde do normálnej prevádzky. Prijímací bafer sa vyprázdni a dáta sa stratia.

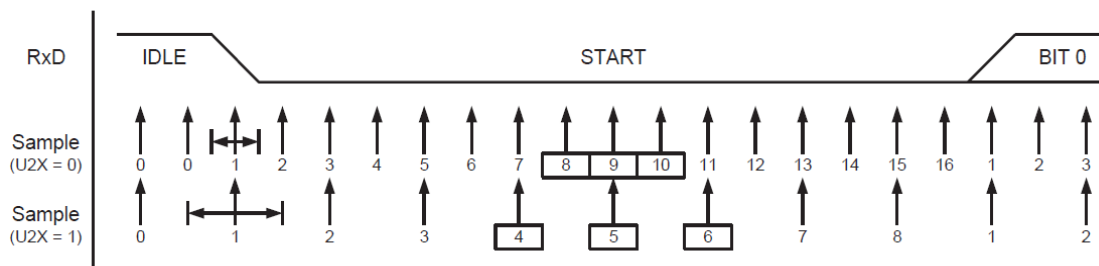
USART- Asynchronný príjem dát

Prijímač USARTu je tvorený dvoma blokmi: Blok obnovy taktovacej frekvencie a blok obnovy prijatých dát.

Synchronizácia prijímača a vysielacza, „fázovanie“

Logika obnovy dát synchronizuje vnútorné hodiny s prichádzajúcim sériovým rámcom. Vid'. nasledujúci obrázok. V normálnom móde je taktovacia frekvencia 16 krát väčšia ako prenosová frekvencia a pri dvojnásobnej prenosovej frekvencii je 8 krát väčšia.

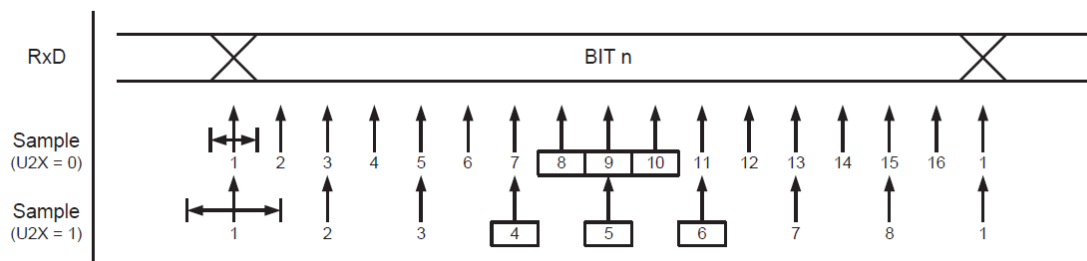
Start Bit Sampling



Prechodom z **IDLE** do **START** na vodiči **RxD** sa synchronizuje vnútorný čítač. Pri každom novom prechode z **IDLE** do **START** sa čítač nuluje. Vnútorná logika potom spracuje vzorky 8, 9 a 10 v normálnom móde a vzorky 4, 5 a 6 pre dvojnásobnú rýchlosť. Ak dve z troch vzoriek sú nuly vyhodnotí sa platný start bit a pokračuje sa v prijímaní ďalších bitov. V opačnom prípade sa takýto pokus vylúči a čaká sa na príchod ďalšieho Start bitu. Synchronizácia sa deje pre každý nový Start bit.

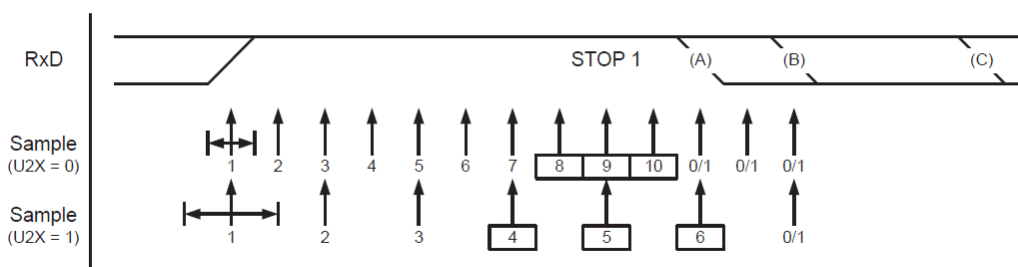
Podobne sa vyhodnocujú aj ostatné prichádzajúce bity.

Sampling of Data and Parity Bit



Stop bit sa vyhodnocuje len prvý. Po prečítaní platného Stop bitu môže nasledovať ďalší Start bit.

Stop Bit Sampling and Next Start Bit Sampling



Viac procesorová komunikácia

stavenie módu multiprocessorová komunikácia *Multi-processor Communication mode* (**MPCMn** bit v registri **UCSRnA**), vyvolá funkciu triedenia prichádzajúcich rámcov do prijímača **USART**-u. Rámce, ktoré neobsahujú informáciu o adrese budú ignorované a nebudú zapísané do prijímacieho bafra. Tento postup podstatne obmedzí počet prijatých rámcov a obslužených v CPU v systéme s viacerými CPU, zariadeniami, pripojenými na tú istú sériovú linku. Nastavenie bitu nemá vplyv na vysielač **USART**-u, avšak vysielačné rámce budú mať po každom iný informačný obsah.

V multiprocessorovej komunikácii možno nastaviť aj rámce s menším počtom dátových bitov, ale tu sa tomu venovať nebudeme. **USART** to podporuje, ale je to zložité.

Ak je prijímač nastavený na príjem rámcov s deviatimi dátovými bitmi, potom deviaty bit (**RxBn8**) je použitý na identifikovanie dátového, resp. adresného rámca. Ak je bit určujúci typ rámca (deviaty bit) jednotkový, **rámec predstavuje adresu**. Ak je bit určujúci typ rámca nulový **rámec predstavuje dáta**. Multi processorová komunikácia umožňuje príjem niekoľkým **SLAVE** zariadeniam z jedného **MASTER** zariadenia. Počas príjmu adresného rámca adresované zariadenia sa odpovedajúco nastavujú. Ak je niektoré **SLAVE** zariadenie adresované, potom bude prijímať nasledujúce dátové rámce a naopak ostatné **SLAVE** zariadenia ich budú ignorovať, do času pokiaľ nebudú adresované.

Prijímače aj vysielač je nastavený na 9-bitový dátový rámec. (**UCSZn = 7**). **MCU** určené do funkcie **MASTER MCU**, nastaví 9-tý dátový bit (**TXB8n = 1**) ak vysiela adresný rámec alebo ho vynuluje ak vysiela dátový rámec (**TXB8n = 0**).

Nasledujúci postup treba dodržať pri *Multi-processor Communication mode*:

1. Všetky **SLAVE** zariadenia sa nastavujú do *Multi-processor Communication mode* (bit **MPCMn=1**, register **UCSRnA**).
2. **Master MCU** vysiela adresný rámec, a všetky **SLAVE** prijímajú a spracujú tento rámec. V **SLAVE** zariadeniach, sa nastaví príznak **RXCn** (*USART Receive Complete*) v registri **UCSRnA**.
3. Každé **SLAVE** zariadenie prečíta obsah registra **UDRn** a určí, či bolo alebo nebolo vybraté - adresované. Ak áno, vynuluje bit **MPCMn** v registri **UCSRnA**, v opačnom prípade čaká na ďalší adresný rámec a ponechá **MPCMn** bit nastavený.
4. Adresované **MCU- zariadenie** bude prijímať dátové rámce až do okamžiku príjmu nového adresného rámca. Ostatné **SLAVE MCU**, ktoré si ponechajú nastavený bit, budú ignorovať dátové rámce.

5. Po prijatí posledného dátového rámca si nastaví adresované zariadenie bit **MPCMn** a čaká na príjem ďalšieho adresného rámca. Proces pokračuje potom bodom 2.

Nepoužívajte inštrukcie *Read-Modify-Write* (**SBI** a **CBI**) na nastavenie alebo nulovanie bitu **MPCMn**. Bit **MPCM** zdieľa tú istú I/O adresu ako príznak **TXCn** a mohol by byť náhodne nulovaný pri použití inštrukcie **SBI** alebo **CBI**.

USART – Popis registrov

USART I/O Data Register n – UDRn

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDRn(Read) UDRn(Write)
	TXB[7:0]								
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	0	0	0	

Registre **UDRn** - *USART Transmit Data Buffer* a *USART Receive Data Buffer* zdieľajú tú istú adresu. Do vysielacieho bafru možno zapísať len vtedy, ak príznak **UDREN** v registri **UCSRnA** je nastavený. Dáta zapísané do **UDRn** počas doby keď je príznak **UDREN** nulový, budú vysielateľom **USART**-u ignorované. Ak bude všetko OK, dáta sa „vyposúvajú cez pin **TxD**“.

Prijímací bafer je dvojúrovňový FIFO. FIFO zmení svoj obsah pokaždé, keď k nemu pristúpime. Ztohto dôvodu nesmieme použiť inštrukciu *read modify write* (**SBI** a **CBI**) pri práci s FIFO. Podobný účinok majú aj inštrukcie testujúce bit (**SBIC** a **SBIS**).

USART Control and Status Register n A – UCSRnA

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

• Bit 7 – RXCn USART Receive Complete

Tento príznak sa nastaví ak sú v prijímacom bafri neprečítané dáta. Vynuluje sa, ak je prijímací bafer prázdny. Ak je prijímač vypnutý, prijímací bafer sa vyprázdni a bit **RXCn** sa vynuluje. Príznak **RXCn** môže generovať prerušenie *Receive Complete interrupt* (viď. Popis bitu **RXCIEn**).

• Bit 6 – TXCn USART Transmit Complete

Tento bit sa nastaví, ak sa celý obsah vysielacieho registra vyposúva a v **UDRn** registri nie sú žiadne dáta pripravené na vysielanie. Bit **TXCn** sa automaticky vynuluje ak sa vykoná prerušenie *transmit complete interrupt*, alebo ho môžeme vynulovať zápisom jednotky na toto miesto. Tento bit generuje prerušenie *Transmit Complete interrupt* (viď. Popis bitu **TXCIEn**).

• Bit 5 – UDREN USART Data Register Empty

Príznak **UDREN** indikuje, či je *transmit buffer* (**UDRn**) pripravený prijať nové dáta. Ak je **UDREN** jednotkové, bafer je prázdny a teda pripravený na to aby sme doň zapísali. Príznak **UDREN** môže generovať prerušenie *Data Register empty Interrupt* (viď. Popis bitu **UDRIEn**). **UDREN** sa po resete nastaví a signalizuje, že vysielateľ je pripravený.

• Bit 4 – FEn Frame Error

Tento bit sa nastaví, ak posledne prijatý znak má chybu rámca - *Frame Error*., T.j. keď je prvý stop bit prijatého rámca nulový. Platnosť tohto bitu sa zruší prečítaním prijímacieho bafru (**UDRn**). Bit **FEn** sa nastaví na nulu ak je prijatý stop bit jednotkový. Pri zápise do registra **UCSRnA** nastavte tento bit na nulu.

• Bit 3 – DORn Data OverRun

Tento bit sa nastaví ak prijaté dáta „pretečú“ - *Data OverRun*. Stav pretečenia sa nastaví, ak je prijímací bafer plný (dva znaky), v posuvnom registri je ďalší znak a na vodiči **RxDn** sa objaví Start bit. Platnosť tohto bitu sa zruší prečítaním prijímacieho bafru (**UDRn**). Pri zápise do registra **UCSRnA** nastavte tento bit na nulu.

• **Bit 2 – PEn Parity Error**

Tento bit sa nastaví, ak posledný prijatý znak ma chybu parity a príznak kontroly parity bol nastavený (**UPMn1** = 1). Platnosť tohto bitu sa zruší prečítaním prijímacieho bafra (**UDRn**). Pri zápise do registra **UCSRnA** nastavte tento bit na nulu.

• **Bit 1 – U2Xn Double the USART Transmission Speed**

Tento bit účinkuje len v asynchrónnom móde. V synchrónnom móde nastavte tento bit na nulu. Nastavenie tohto bitu redukuje deliaci pomer zo 16 na 8.

• **Bit 0 – MPCMn Multi-processor Communication Mode**

Tento bit povoľuje – nastavuje *Multi-processor Communication mode*. Ak ho nastavíme na jedna, všetky prichádzajúce rámce, prijaté prijímačom budú ignorované, ak nebudú mať nastavenú informáciu o adrese. Na vysielač tento bit nepôsobí.

USART Control and Status Register n B – UCSRnB

Bit	7	6	5	4	3	2	1	0	
	RXCIE _n	TXCIE _n	UDRIE _n	RXEN _n	TXEN _n	UCSZ _{n2}	RXB8 _n	TXB8 _n	UCSRnB
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	

• **Bit 7 – RXCIE_n RX Complete Interrupt Enable**

Nastavenie tohto bitu na jednotku povolí prerušenie vyvolané nastavením bitu **RXC_n** (predpokladáme, že **GI** je povolené).

• **Bit 6 – TXCIE_n TX Complete Interrupt Enable**

Nastavenie tohto bitu na jednotku povolí prerušenie vyvolané nastavením bitu **TXC_n** (predpokladáme, že **GI** je povolené).

• **Bit 5 – UDRIE_n USART Data Register Empty Interrupt Enable**

Nastavenie tohto bitu na jednotku povolí prerušenie vyvolané nastavením bitu **UDRE_n** (predpokladáme, že **GI** je povolené).

• **Bit 4 – RXEN_n Receiver Enable**

Nastavenie tohto bitu na jednotku zapne prijímač **USART**-u. Pin **RXD** prejde pod riadenie prijímača. Vypnutie prijímača uvoľní – vyprázdni prijímací bafer a odstavi príznaky **FE**, **DOR** a **PE**.

• **Bit 3 – TXEN_n Transmitter Enable**

Nastavenie tohto bitu na jednotku zapne vysielač **USART**-u. Pin **TXD** prejde pod riadenie vysielača. Vypnutie vysielača uvoľní – nastavenie bitu na nulu zareaguje až po odvysielaní všetkých dát určených na vysielanie.

• **Bit 2 – UCSZ_{n2} Character Size**

Bit **UCSZ₂** spolu s **UCSZ_{1:0}** bitmi v **UCSRnC** nastavuje počet dátových bitov (Character Size) pre rámec (frame) prijímača a vysielača.

• **Bit 1 – RXB8_n Receive Data Bit 8**

RXB8_n je deviaty dátový bit prijatý prijímačom pri deväť bitovom rámci. Musíme ho čítať skorej ako prečítame **UDRn** register.

• **Bit 0 – TXB8_n: Transmit Data Bit 8**

TXB8_n je deviaty dátový bit vysielačom pri deväť bitovom rámci. Musíme ho nastaviť skorej ako zapíšeme do **UDRn** registra.

USART Control and Status Register n C – UCSRnC

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

• Bit 7:6 – UMSELn1:0 USART Mode Select

Tieto bity prepínajú medzi asynchrónnou a synchrónnou prevádzkou, vid' tabuľka.

UMSELn Bits Settings

UMSELn1	UMSELn0	Mode
0	0	Asynchronous USART
0	1	Synchronous USART
1	0	(Reserved)
1	1	Master SPI (MSPIM) ⁽¹⁾

• Bit 5:4 – UPM1:0 Parity Mode

Tieto bity zapínajú a nastavujú typ kontrolovanej parity. Ak je kontrola zapnutá, vysielač automaticky generuje a pošle odpovedajúci paritný bit. Prijímač tiež automaticky generuje odpovedajúci paritný bit prichádzajúceho rámca a porovná výsledky v zhode s nastavením bitu **UPM0**. Pri nezhode bit **UPEn** v registri **UCSRnA** sa nastaví.

UPMn Bits Settings

UPMn1	UPMn0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity

• Bit 3 – USBSn Stop Bit Select

Tento bit volí počet stop bitov pri vysielaní. Na prijímač nemá nastavenie tohto bitu vplyv. Vid' Tabuľka.

USBS Bit Settings

USBSn	Stop Bit(s)
0	1-bit
1	2-bit

• Bit 2:1 – UCSZn1:0 Character Size

Bity **UCSZn1:0** spolu s bitom **UCSZn2** v **UCSRnB** nastavujú počet dátových bitov rámca vysielača a prijímača. Vid' Tabuľka.

UCSZn Bits Settings

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

• Bit 0 – UCPOLn Clock Polarity

Tento bit je použitý len pre synchrónne módy – my sa venujeme len asynchrónnym módom. Nastav na nulu.

USART Baud Rate Registers – UBRRnL and UBRRnH

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/W	R	R	R	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	

- **Bit 15:12- Reserved**

Tieto bity sú rezervované pre budúce použitie a treba ich nastaviť na nulu pri zápise do **UBRRnH**.

- **Bit 11:0 – UBRRn11:0 USART Baud Rate Register**

Toto je 12-bitový register, ktorý obsahuje prenosovú rýchlosť **USART**. Do **UBRRnH** zapisujeme štyri najvýznamnejšie bity a do **UBRRnL** zapisujeme osem menej významných bitov prenosovej rýchlosti **USART**. Zápis do **UBRRnL** okamžite pozmení obsah preddeliča podľa novej prenosovej rýchlosti.

Príklady nastavenia prenosovej rýchlosti

Pre štandardné frekvencie kryštálu a rezonátora sú formou tabuľky uvedené najpoužívanejšie prenosové rýchlosti asynchrónnej prevádzky nastavované pomocou **UBRRn**. Hodnoty **UBRRn**, ktoré poskytujú prenosové rýchlosti s chybou menšou ako 0.5%, sú v tabuľke zapísané hrubo. Väčšie chyby sú akceptovateľné, ale prijímač bude mať menšiu odolnosť voči šumom. Hlavne pre väčšie sériové rámce (viď. “Asynchronous Operational Range” str. 157). Chyba je počítaná pomocou rovnice:

$$\text{Error}[\%] = \left(\frac{\text{BaudRate}_{\text{Closest Match}}}{\text{BaudRate}} - 1 \right) \cdot 100\%$$

Table 68, 69,70, ... Examples of UBRR Settings for Commonly Used Oscillator Frequencies

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

Baud Rate (bps)	$f_{\text{osc}} = 8.0000 \text{ MHz}$				$f_{\text{osc}} = 11.0592 \text{ MHz}$				$f_{\text{osc}} = 14.7456 \text{ MHz}$			
	$U2Xn = 0$		$U2Xn = 1$		$U2Xn = 0$		$U2Xn = 1$		$U2Xn = 0$		$U2Xn = 1$	
	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error	UBRRn	Error
2400	207	0.2%	416	-0.1%	287	0.0%	575	0.0%	383	0.0%	767	0.0%
4800	103	0.2%	207	0.2%	143	0.0%	287	0.0%	191	0.0%	383	0.0%
9600	51	0.2%	103	0.2%	71	0.0%	143	0.0%	95	0.0%	191	0.0%
14.4k	34	-0.8%	68	0.6%	47	0.0%	95	0.0%	63	0.0%	127	0.0%
19.2k	25	0.2%	51	0.2%	35	0.0%	71	0.0%	47	0.0%	95	0.0%
28.8k	16	2.1%	34	-0.8%	23	0.0%	47	0.0%	31	0.0%	63	0.0%
38.4k	12	0.2%	25	0.2%	17	0.0%	35	0.0%	23	0.0%	47	0.0%
57.6k	8	-3.5%	16	2.1%	11	0.0%	23	0.0%	15	0.0%	31	0.0%
76.8k	6	-7.0%	12	0.2%	8	0.0%	17	0.0%	11	0.0%	23	0.0%
115.2k	3	8.5%	8	-3.5%	5	0.0%	11	0.0%	7	0.0%	15	0.0%
230.4k	1	8.5%	3	8.5%	2	0.0%	5	0.0%	3	0.0%	7	0.0%
250k	1	0.0%	3	0.0%	2	-7.8%	5	-7.8%	3	-7.8%	6	5.3%
0.5M	0	0.0%	1	0.0%	-	-	2	-7.8%	1	-7.8%	3	-7.8%
1M	-	-	0	0.0%	-	-	-	-	0	-7.8%	1	-7.8%
Max. ⁽¹⁾	0.5 Mbps		1 Mbps		691.2 kbps		1.3824 Mbps		921.6 kbps		1.8432 Mbps	

1. UBRRn = 0, Error = 0.0%