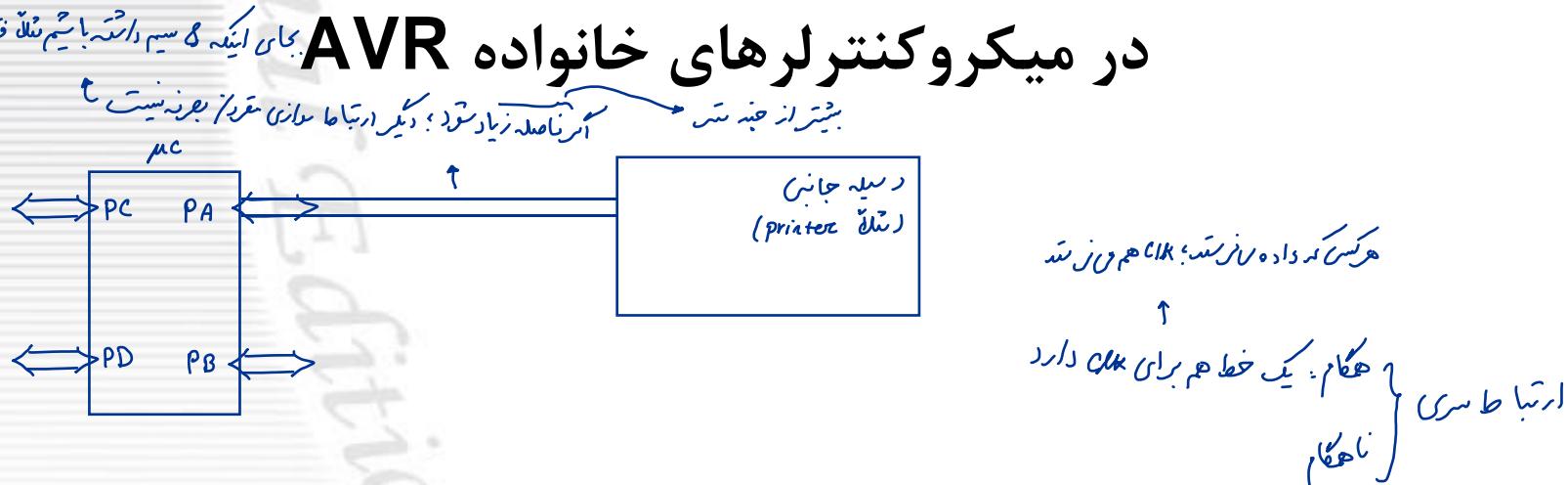




Universal Synchronous/Asynchronous Receiver/Transmitter USART





فهرست مطالب

مطالبی که در این فصل با آنها آشنا خواهیم شد:
• ارتباط USART



واحد USART

- فرستنده و گیرنده سریال همگام و ناهمگام همه منظوره (USART) یک وسیله ارتباطی سریال بسیار انعطاف‌پذیر برای برقراری ارتباط همگام یا ناهمگام مابین واحد پردازشگر مرکزی و دستگاه‌های جانبی است
- ارتباط سریال همگام : علاوه بر خطوط ارسال و دریافت داده، خط ساعت هم لازم است (سیگнал ساعت روی پایه XCK منتقل می‌شود)
- ارتباط سریال ناهمگام : Clock حذف می‌شود و با تنظیم یکسان سرعت انتقال داده در فرستنده و گیرنده، فقط یک خط برای ارسال و یک خط برای دریافت لازم است (ارسال روی پایه TxD و دریافت روی پایه RxD انجام می‌شود)



مشخصات اصلی USART

مشخصات اصلی USART:

- عملیات کاملاً دوطرفه (ثبت‌های دریافت و ارسال سریال مستقل)
- عملیات همگام یا ناهمگام
- عملیات همگام اصلی راهبر یا پیرو
Master Slave
- مولد نرخ باد با دقت بالا
- پشتیبانی فریم‌های سریال با ۵، ۶، ۷، ۸ یا ۹ بیت داده و ۱ یا ۲ بیت توقف
- تولید توازن فرد یا زوج و بررسی کردن توازن توسط سخت‌افزار
- تشخیص خطای سرروی (به معنی روی هم افتادگی داده‌های دریافتی جدید روی داده‌های دریافتی قبلی است که منجر به از بین رفتن داده قبلی می‌شود)

overrun error



مشخصات اصلی USART

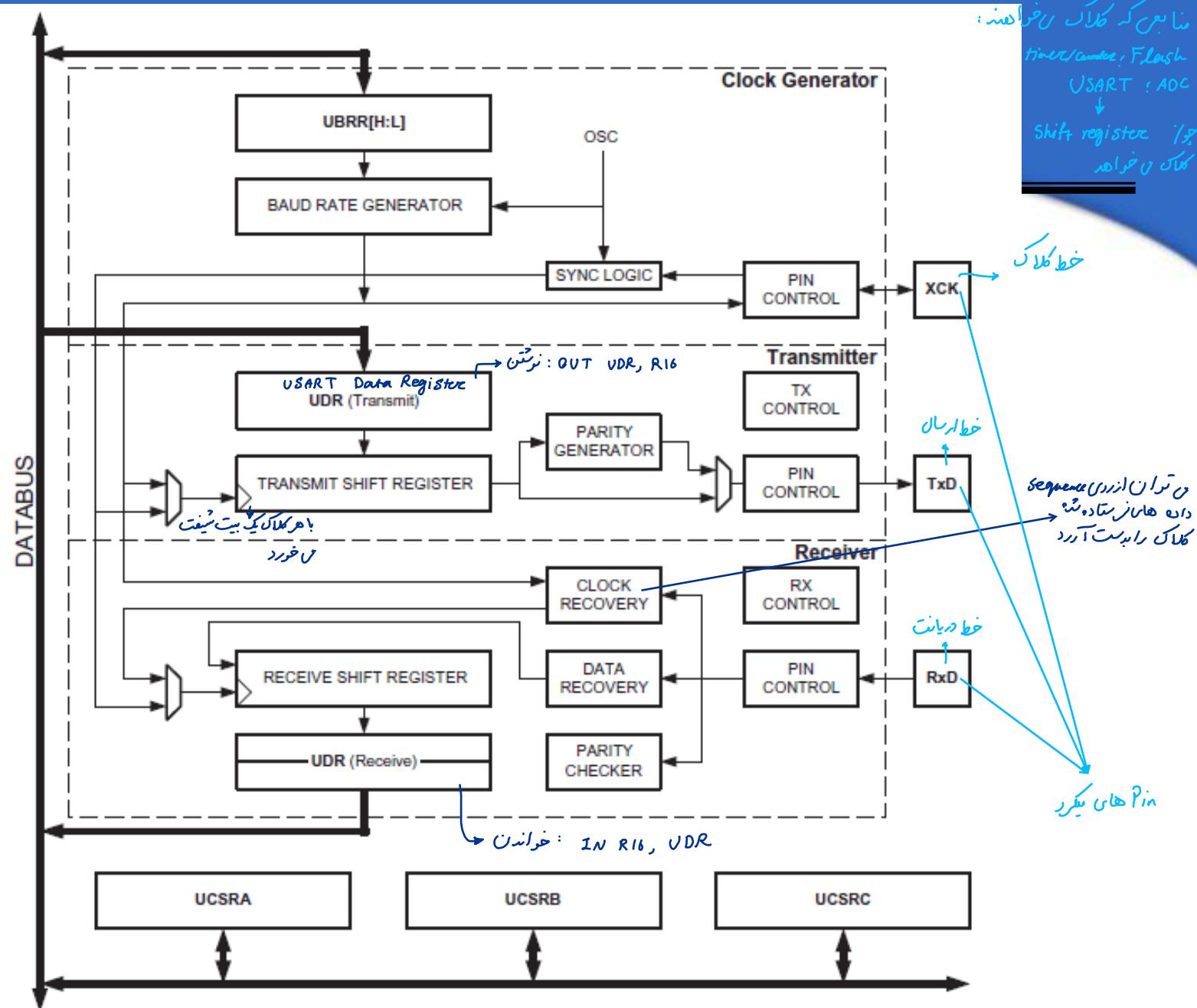
• مشخصات اصلی USART (ادامه)

• تشخیص خطای فریم

- فیلتر کردن نویز شامل تشخیص بیت شروع اشتباه و فیلتر پائین‌گذر رقمه‌ی TX (Transmit Complete)
- سه وقفه جداگانه شامل وقفه ناشی از اتمام ارسال (TX), وقفه ناشی از خالی بودن ثبات داده ارسال (UDRE) و وقفه ناشی از اتمام دریافت (RX) (Receiver Complete)
- مود ارتباط چند پردازنده‌ای ← Master Slave
- مود ارتباط ناهمگام با سرعت دوبرا بر



نایاب کر کار رخواهی:
timer/counter, Flash
USART : ADC
↓
Shift register ۱۶
کار رخواهی





تولید سیگنال ساعت

- واحد تولید ساعت، ساعت پایه برای گیرنده و فرستنده را تولید می‌نماید.
- ماژول USART چهار حالت تولید ساعت را پشتیبانی می‌نماید که شامل حالت عادی ناهمگام، حالت ناهمگام با سرعت دوباره، حالت همگام راهبر و حالت هنگام پیرو می‌باشد
- بیت UMSEL در ثبات کنترل و وضعیت UCSRC، وظیفه انتخاب بین حالات همگام و ناهمگام را بر عهده دارد.
- بیت U2X، در ثبات UCSRA، سرعت دوباره برای حالت ناهمگام را کنترل می‌کند
- هنگام استفاده از حالت همگام ($UMSEL=1$)، ثبات جهت داده برای پایه XCK (ثبات DDR-XCK) تعیین کننده‌ی منبع ساعت داخلی یا خارجی است.
- پایه XCK تنها زمان استفاده از حالت همگام فعال است



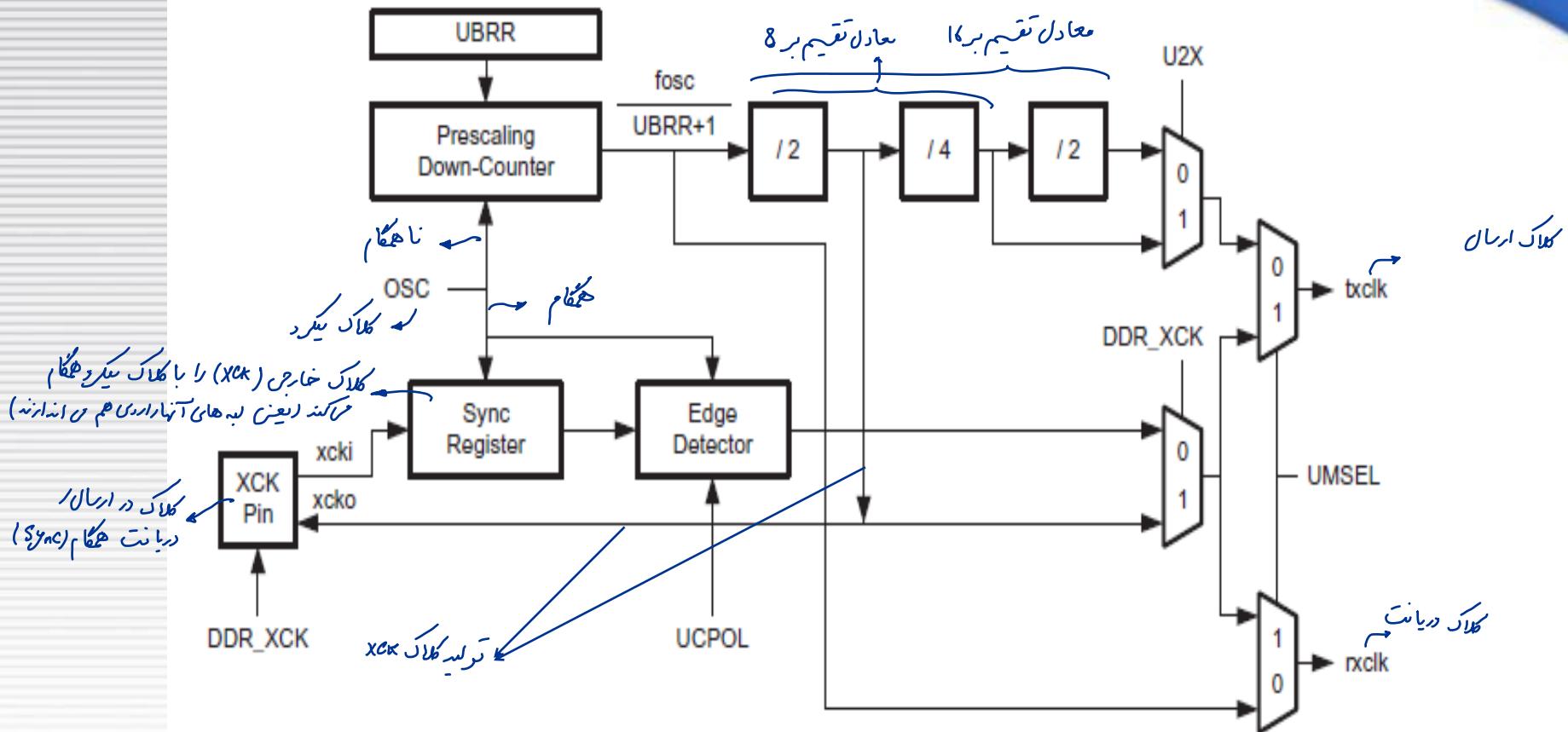
توصیف ثبات‌ها

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDR (Read)
	TXB[7:0]								UDR (Write)
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	
Frame busy end, Start of frame busy before framing error	Framing Error								واده دریافت جدید را داده درینه تبلیغ نوشته شده است
Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	
Parity error									
Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
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	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
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	

* خواندن/خطای خواندن باید قبل از خواندن UDR دستورات باشد؛ اگر UDR خوانده بشود؛ یعنی خطای خواندن



نمودار بلوکی واحد مولد ساعت





نمودار بلوکی واحد مولد ساعت

توصیف سیگنال‌ها:

- txclk: ساعت ارسال که یک سیگنال داخلی است.
- rxclk: ساعت گیرنده که یک سیگنال داخلی است
- xcki: ورودی از پایه XCK. برای عملیات راهبر همگام بکار می‌رود
- xcko: خروجی ساعت به پایه XCK (ساعت داخلی). برای عملیات راهبر همگام بکار می‌رود
- fosc: فرکانس پایه XTAL (همان ساعت سیستم)



تولید سیگنال ساعت ورودی-مولد نرخ باد

- ثبات UBRR و شمارنده نزولی متصل به آن، به عنوان یک پیش تقسیم کننده فرکانس یا مولد نرخ باد قابل برنامه ریزی عمل می کند
- هر وقت این شمارنده نزولی صفر شود، یک سیگنال ساعت تولید می شود
- فرستنده بر اساس مدد خروجی ساعت مولد نرخ باد را برابر ۲، ۸ و یا ۱۶ تقسیم می کند
- خروجی مولد نرخ باد مستقیماً توسط واحد ساعت گیرنده و واحد بازیابی داده مورد استفاده قرار می گیرد.



توصیف ثبات‌ها

UBRR = USART Band Rate Register

Bit	15	14	13	12	11	10	9	8	UBRRH	UBRRL
	URSEL	-	-	-	UBRR[11:8]					
					UBRR[7:0]					
	7	6	5	4	3	2	1	0		
Read/Write	R/W	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		



روابطی برای محاسبه تنظیمات ثبات نرخ باد

معادله محاسبه مقدار UBRR	معادله محاسبه نرخ باد	مد عملیاتی
$UBRR = \frac{f_{osc}}{16BAUD} - 1$	$BAUD = \frac{f_{osc}}{16(UBRR+1)}$	مد عادی ناهمگام (U2X = 0)
$UBRR = \frac{f_{osc}}{8BAUD} - 1$	$BAUD = \frac{f_{osc}}{8(UBRR+1)}$	مد سرعت دوباره ناهمگام (U2X = 1)
$UBRR = \frac{f_{osc}}{2BAUD} - 1$	$BAUD = \frac{f_{osc}}{2(UBRR+1)}$	مد عصر هنگام راهبرد Synchronous Master Mode

توجه: ۱- نرخ باد بصورت نرخ انتقال اطلاعات بر حسب بیت بر ثانیه تعریف می‌شود (bps).

BAUD: نرخ باد (بر حسب بیت بر ثانیه (bps))
fosc: فرکانس ساعت نوسان‌ساز سیستم

UBRR: محتوای ثبات‌های UBRRH و UBRRL (۰ - ۴۰۹۵).



عملیات با سرعت مضاعف (U2X)

- یک کردن این بیت موجب کاهش مقسوم علیه تقسیم کننده نرخ باد از ۱۶ به ۸ می شود، این کار عمل نرخ باد را برای ارتباطات ناهمگام دو برابر می کند
- یک کردن این بیت صرفا بر عملیات ناهمگام تاثیر دارد؛ این بیت باید در عملیات همگام صفر شود
- در این مورد گیرنده تنها از نصف تعداد نمونه ها برای بازیابی ساعت و نمونه برداری داده ها استفاده می نماید
- وقتی که از این حالت استفاده می کنیم، ساعت سیستم و نرخ باد دقیقتری مورد نیاز است



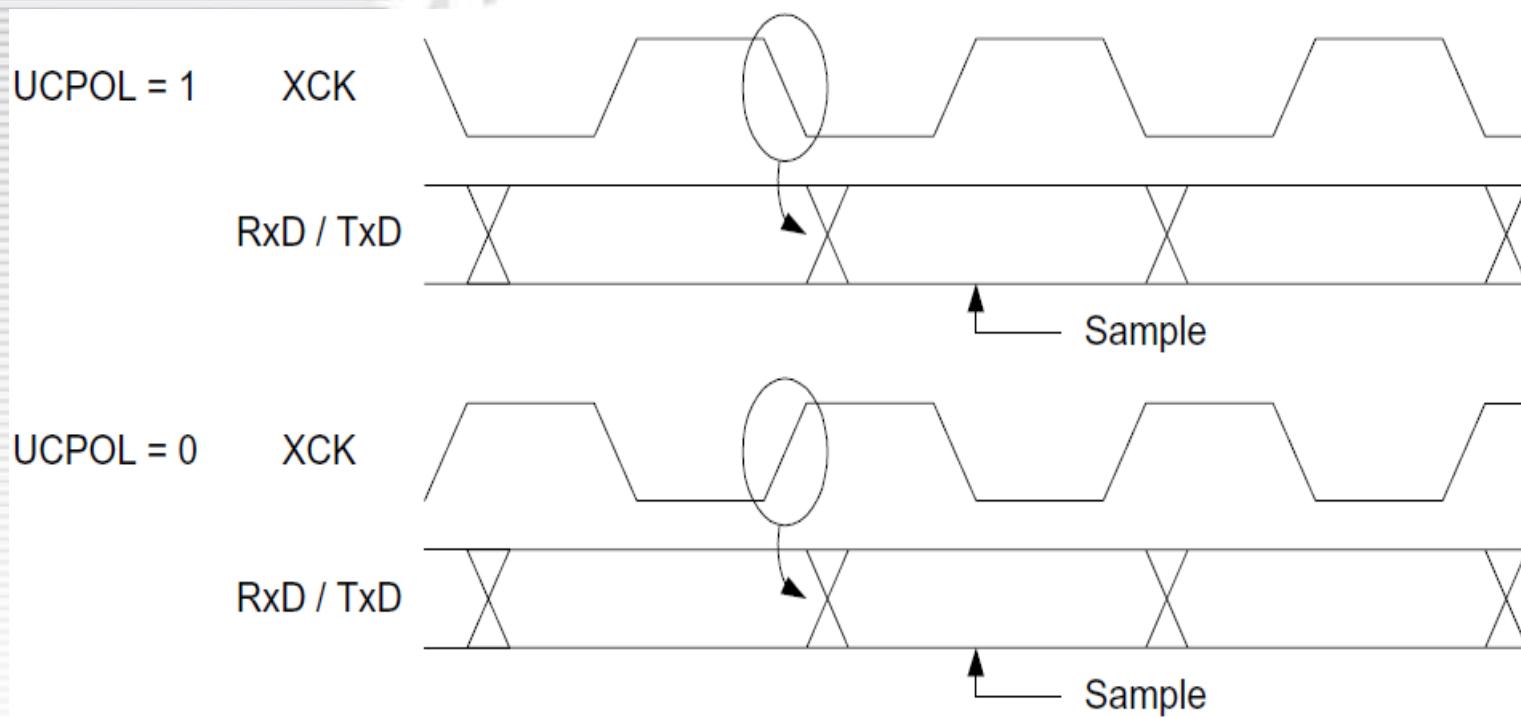
ساعت خارجی

- همگام‌سازی از طریق ساعت خارجی، توسط حالت‌های عملیاتی پیرو همگام مورد استفاده قرار می‌گیرد
- ورودی ساعت خارجی از پایه XCK توسط یک ثبات همگام‌سازی نمونه‌برداری می‌شود تا احتمال ناپایداری به حداقل برسد
- خروجی این ثبات قبل از بکارگیری باید از یک آشکارساز لبه عبور کند؛ که تاخیری به اندازه دوپریود ساعت CPU ایجاد می‌کند (بیشینه فرکانس ساعت خارجی)
- دقت f_{osc} به پایداری منبع ساعت سیستم بستگی دارد؛ بنابر این، فرکانس ساعت خارجی را کمی از مقدار بدست امده از رابطه فوق کمتر می‌گیرند



عملیات ساعت همگام

- در حالت همگام (1)، پایه XCK می‌تواند به عنوان ورودی ساعت و یا خروجی ساعت مورد استفاده قرار گیرد

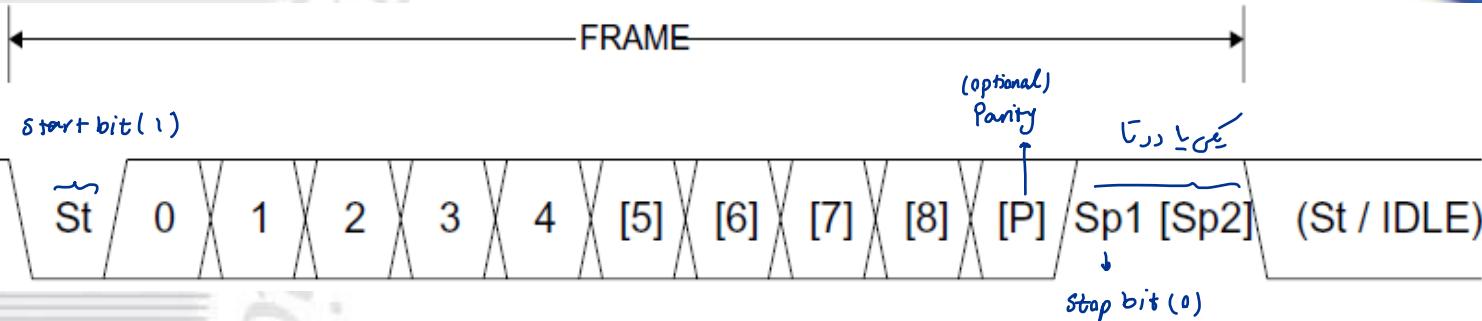




قالب فریم‌ها

- فریم یا قالب سریال: یک کاراکتر از بیت‌های داده به همراه بیت‌های همگام‌سازی (همچون بیت‌های شروع و پایان) و نیز بیت توازن که وجود آن اختیاری بوده و برای تشخیص خطاب کار می‌رود
- همه ۳۰ ترکیب ممکن از موارد زیر توسط USART به عنوان قالب‌های صحیح فریم پذیرفته می‌شوند:
 - یک بیت شروع
 - پنج، شش، هفت، و یا نه بیت داده
 - هیچ و یا یک بیت توازن فرد یا زوج
 - یک یا دو بیت پایان
- در آغاز فریم، بیت شروع و به دنبال آن کم‌ارزش‌ترین بیت داده، بعد از آن بیت‌های داده بعدی و با بالاترین بیت، خاتمه می‌یابند

قالب فریم ها



- قالب های فریم مورد استفاده توسط USART با استفاده از بیت های 0 **UCSZ2:0**، **USBS** و بیت های **SPM1:0** در ثبات های **UCSRB** و **UCSRC** تنظیم می شوند (گیرنده و فرستنده از تنظیمات یکسانی استفاده می کنند)
- بیت های اندازه کاراکتر USART، تعداد بیت های داده در هر فریم را مشخص می کنند
- انتخاب اینکه یک یا دو بیت توقف داشته باشیم توسط بیت **USBS** تعیین می شود
- یک خطای قاب تنها زمانی آشکار می شود که اولین بیت توقف صفر باشد



محاسبه‌ی بیت توازن

بیت توازن با XOR کردن همه بیت‌های داده بدست می‌آید

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

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

بیت توازن با استفاده از توازن زوج
بیت توازن با استفاده از توازن فرد
بیت داده nام از کاراکتر مورد نظر

P_{even}
 P_{odd}
 d_n

- در صورتی که از امکان توازن استفاده شود، بیت توازن بین آخرین بیت داده و اولین بیت توقف واقع خواهد شد.



برنامه‌ریزی اولیه‌ی USART

در مورد عملیات USART وقفه‌گرا، پرچم وقفه سراسری باید هنگام برنامه‌ریزی اولیه صفر شود (تمامی وقفه‌ها غیر فعال شوند)

- اول باید مطمئن شد هیچ انتقال در حال انجامی در طی دوره تغییر ثبات‌ها وجود ندارد
- پرچم TXC برای بررسی تکمیل ارسال و پرچم RXC برای بررسی وجود داده‌های خوانده نشده در بافر گیرنده استفاده می‌شود (پرچم TXC قبل از هر انتقال صفر شود)
- هنگامی که تابع در ثبات UCSRC چیزی را می‌نویسد، بیت URSEL (با ارزش ترین بیت)، به دلیل به اشتراک گذاشته شدن مکان I/O توسط UBRRH و UCSRC، باید یک شود



نمونه کد به زبان اسambilی

Assembly Code Example

USART_Init:

; Set baud rate

out UBRRH, r17

out UBRRL, r16

; Enable receiver and transmitter

ldi r16, (1<<RXEN) | (1<<TXEN)

out UCSRB, r16

; Set frame format: 8data, 2stop bit

ldi r16, (1<<URSEL) | (1<<USBS) | (3<<UCSZ0) کعن است اسambil اجراند

⁰¹¹

out UCSRC, r16

ret



نمونه کد به زبان C

C Code Example

```
#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 */
    UBRRH = (unsigned char)(baud>>8);
    UBRLR = (unsigned char)baud;
/* Enable receiver and transmitter */
    UCSRB = (1<<RXEN)|(1<<TXEN);
/* Set frame format: 8data, 2stop bit */
    UCSRC = (1<<URSEL)|(1<<USBS)|(3<<UCSZ0);
}
```



انتقال داده‌ها-فرستنده USART

- فرستنده USART به واسطه یک کردن بیت TXEN در ثبات UCSRB فعال می‌شود
- بعد از فعال شدن فرستنده، کارکرد عادی پایه TXD به عنوان درگاه، تغییر یافته و به خروجی سریال فرستنده تبدیل می‌شود
- نرخ باد، حالت عملیات و قالب فریم، باید یک بار قبل از هرگونه انتقالی تنظیم شود
- در صورتی که عملیات همگام مورد استفاده قرار گیرد، عملکرد عادی پایه XCK ملغی شده و ساعت موجود بر روی پایه XCK به عنوان ساعت ارسال مورد استفاده قرار خواهد گرفت



توصیف ثبات‌ها

Bit	7	6	5	4	3	2	1	0	
	RXB[7:0]								UDR (Read)
	TXB[7:0]								UDR (Write)
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	پس از ارسال در تابعی تناظری: پس از ارسال خارجی
Bit	Receive Complete	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	
Bit	Receive complete Interrupt Enable	7	6	5	4	3	2	1	0
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
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	رتفه از نزع یک
Bit	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
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	



توصیف ثبات‌ها

Bit	15	14	13	12	11	10	9	8	UBRRH
	URSEL	-	-	-	UBRR[11:8]				UBRRL
					UBRR[7:0]				
Read/Write	R/W	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	



ارسال فریم‌هایی با ۵ تا ۸ بیت داده

- ارسال داده‌ها با بارگذاری بافر **ارسال**، با داده‌ای که قرار است ارسال شود آغاز می‌گردد
- هنگامی که ثبات انتقال آماده ارسال یک فریم جدید باشد، داده‌های بافر شده در بافر ارسال به ثبات ارسال منتقل می‌شوند
- ثبات ارسال هنگامی که در حالت بیکاری باشد و یا بلافاصله بعد از ارسال آخرین بیت توقف فریم قبلی، با داده‌های جدید بارگذاری می‌شود
- هنگامی که ثبات ارسال با داده جدید بارگذاری شود، یک فریم کامل را بسته به حالت عملیاتی مورد استفاده با نرخ مشخص شده توسط ثبات مربوط به نرخ باد، بیت U2X یا توسط XCK ارسال می‌کند



نمونه کد به زبان اسembلی و C

Assembly Code Example

USART_Transmit:

```
; Wait for empty transmit buffer
sbis UCSRA,UDRE          آنکه برد بعنه داده تبلیغ ارسال شده
rjmp USART_Transmit
; Put data (r16) into buffer, sends the data
out UDR,r16
ret
```

C Code Example

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



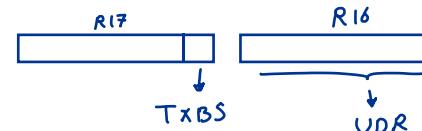
ارسال فریم‌هایی با ۹ بیت داده

- در صورتی که طول کاراکتر ۹ بیت باشد، باید قبل از اینکه بایت کم ارزش در UDR نوشته شود، نهمین بیت را در بیت TXB8 واقع در UCSRB بنویسیم

Assembly Code Example

```
USART_Transmit:
    ; Wait for empty transmit buffer
    sbis UCSRA,UDRE
    rjmp USART_Transmit
    ; Copy 9th bit from r17 to TXB8
    ldi r17, TXB8
    cbi UCSRB,TXB8
    sbrc r17,0
    sbi UCSRB,TXB8
    ; Put LSB data (r16) into buffer, sends the data
    out UDR,r16
    ret
```

(l: Nice



C Code Example

```
void USART_Transmit( unsigned int data )
{
    /* Wait for empty transmit buffer */
    while ( !( UCSRA & (1<<UDRE)) )

    /* Copy 9th bit to TXB8 */
    UCSRB &= ~(1<<TXB8);
    if ( data & 0x0100 )
        UCSRB |= (1<<TXB8);
    /* Put data into buffer, sends the data */
    UDR = data;
}
```



دریافت داده‌ها-گیرنده USART

- گیرنده USART به واسطه یک کردن بیت RXEN (فعال‌سازی گیرنده) در ثبات **UCSRB** فعال می‌شود
- بعد از فعال شدن گیرنده، عملکرد عادی پایه RXD به عنوان درگاه، تغییر یافته و به ورودی سریال فرستنده تبدیل می‌شود
- نرخ باد، حالت عملیات و قالب فریم، باید یک بار قبل از هرگونه دریافت سریالی تنظیم شوند.
- در حالت همگام ساعت موجود بر روی پایه XCK به عنوان ساعت انتقال مورد استفاده قرار خواهد گرفت



دریافت فریم‌هایی با ۵ تا ۸ بیت داده

- دریافت داده‌ها با تشخیص یک بیت شروع، آغاز می‌گردد
- هر بیتی که بعد از بیت شروع قرار گیرد، با نرخ باد تعیین شده توسط ساعت XCK نمونه‌برداری می‌شود و تا دریافت بیت پایان، در ثبات انتقال دریافت، جابجا می‌شود
- بیت پایانی دوم توسط گیرنده نادیده گرفته می‌شود
- هنگامی که اولین بیت پایان دریافت شد، یعنی یک فریم سریال کامل در ثبات جابجایی گیرنده آماده شد، محتوای ثبات جابجایی به بافر دریافت انتقال می‌یابد
- بافر دریافت سپس با خواندن محل **UDR** خوانده می‌شود



نمونه کد به زبان اسembلی و C

Assembly Code Example

USART_Receive:

```
; Wait for data to be received
sbis UCSRA, RXC
rjmp USART_Receive
; Get and return received data from buffer
in r16, UDR
ret
```

C Code Example

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



دربیافت فریم‌هایی با ۹ بیت داده

- در این صورت قبل از خوانده شدن بایت پائین از UDR، باید نهمین بیت در بیت موجود در UCSRB خوانده شود 
- همچنین برای پرچم‌های وضعیت PE، DOR و FE، این قانون اعمال می‌شود
- ابتدا پرچم‌های وضعیت از ثبات UCSRA و سپس داده از ثبات UDR خوانده شود



نمونه کد به زبان اسambilی

Assembly Code Example

USART_Receive:

; Wait for data to be received

sbis UCSRA, RXC

rjmp USART_Receive

; Get status and 9th bit, then data from buffer

in r18, UCSRA خواندن/خطاها باید قبل از فرآنده UDR را بازدید نمایند

in r17, UCSRB بیت نهم را بخوانند

in r16, UDR

; If error, return -1

andi r18,(1<<FE)|(1<<DOR)|(1<<PE) } *
breq USART_ReceiveNoError اگر خطای رخداده باشد req انجام نشود (zero flag)

ldi r17, HIGH(-1) } عدد -1 را مانند بسته بخواهد

ldi r16, LOW(-1)

USART_ReceiveNoError:

; Filter the 9th bit, then return

lsl r17 شیفت دادیم تا RXB8 به خانه اول r17 منتقل شود

andi r17, 0x01

ret



نمونه کد به زبان C

C Code Example

```
unsigned int USART_Receive( void )
{
    unsigned char status, resh, resl;
    /* Wait for data to be received */
    while ( !(UCSRA & (1<<RXC)) )

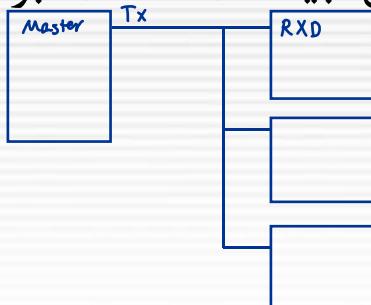
    /* Get status and 9th bit, then data */
    /* from buffer */
    status = UCSRA;
    resh = UCSR_B;
    resl = UDR;
    /* If error, return -1 */
    if ( status & (1<<FE)|(1<<DOR)|(1<<PE) )
        return -1;
    /* Filter the 9th bit, then return */
    resh = (resh >> 1) & 0x01;
    return ((resh << 8) | resl);
}
```



مد ارتباطی چند پردازنده‌ای

Multi Processor Communication Mode

- یک کردن بیت **MPCM** (مد ارتباطی چند پردازنده‌ای) در ثبات UCSRA، یک قابلیت فیلتر کردن فریم‌های دریافتی توسط گیرنده USART را فعال می‌کند
- در اینصورت فریم‌هایی که حاوی اطلاعات آدرس نیستند، چشم‌پوشی خواهند شد
- این امر، تعداد فریم‌های وارد به میکروکنترلر را، در سیستمی که در آن چندین میکروکنترلر از طریق گذرگاه سریال با هم در ارتباط هستند، کاهش می‌دهد
- هنگام تنظیم برای دریافت فریم‌هایی با ۹ بیت داده، نهمین بیت (Rx8) برای شناسایی فریم‌های آدرس و داده بکار خواهد رفت





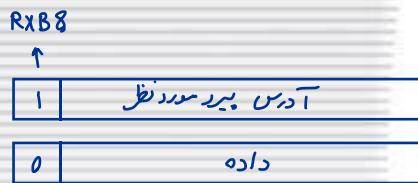
مد ارتباطی چند پردازنده‌ای

- مد ارتباطی چند پردازنده‌ای این قابلیت را فراهم می‌سازد که چندین میکروکنترلر پیرو از یک میکروکنترلر راهبر، داده دریافت کنند
- این عمل ابتدا با دیکد کردن یک فریم آدرس انجام می‌شود؛ که مشخص می‌کند کدام میکروکنترلر آدرس دهی شده
- در صورتی که یکی از میکروکنترلرهای پیرو آدرس دهی شود، آن میکروکنترلر فریم‌های داده بعدی را بطور عادی دریافت خواهد کرد و سایر میکروکنترلرهای پیرو، فریم‌های داده را تا آمدن یک فریم آدرس دیگر، نادیده خواهند گرفت



استفاده از MPCM

- یک میکروکنترلر راهبر، از یک قالب فریم با کاراکتر ۹ بیتی استفاده می‌کند. **نهمین** بیت، هنگام ارسال فریم آدرس باید یک و هنگام ارسال فریم داده، باید صفر شود. در این حالت میکروکنترلرهای پیرو باید از فریم‌هایی با کاراکترهای ۹ بیتی استفاده کنند



* بر است ارسن داده؛ تعداد داده ها باشد (التبه زدن ندارد)

- رویه‌ی تبادل داده‌ها، در حالت ارتباطی چند پردازنده‌ای :

 1. تمامی میکروکنترلرها در حالت ارتباطی چند پردازنده‌ای میکروکنترلر راهبر یک فریم آدرس ارسال، همه پیروها، آن را دریافت و محتوای آن را می‌خوانند. در میکروکنترلرهای پیرو پرچم RXC یک خواهد شد
 2. هر میکروکنترلر پیرو ثبات UDR را می‌خواند تا بفهمد، انتخاب شده است یا نه. اگر انتخاب شده باشد، بیت MPCM را در UCSRA صفر؛ واگرنه منتظر بایت آدرس بعدی مانده و تنظیمات MPCM را حفظ می‌کند



استفاده از MPCM

- رویه‌ی تبادل داده‌ها، در حالت ارتباطی چند پردازنده‌ای (ادامه):
 4. میکروکنترلر آدرس دهی شده تمامی فریم‌های داده را تا دریافت یک فریم آدرس جدید، دریافت خواهد کرد. سایر میکروکنترلرهای پیرو که هنوز بیت MPCM را یک نگه داشته‌اند، فریم‌های داده را نادیده خواهند گرفت
 5. هنگام دریافت آخرین فریم داده توسط میکروکنترلر آدرس دهی شده، میکروکنترلر آدرس دهی شده، بیت MPCM را یک کرده و منتظر یک فریم آدرس جدید از میکروکنترلر راهبر می‌ماند. فرآیند از گام ۲ تکرار می‌شود



دسترسی به ثبات‌های UCSRC و UBRRH

- ثبات UBRRH همان مکان I/O ثبات UCSRC را استفاده می‌نماید
- دسترسی نوشتمن:
- هنگام نوشتمن در این مکان I/O، بیت با ارزش مقدار نوشتمن، تعیین می‌کند که در کدام یک از ثبات‌های USART یا URSEL، باید نوشتمن شود
- در صورتی که این بیت صفر باشد، مقدار UBRRH و در صورتی که این بیت یک باشد، بروزرسانی خواهد شد



نمونه کد به زبان اسembly و C

Assembly Code Example

```
...  
; Set UBRRH to 2  
ldi r16,0x02 بیت برآورده شده است  
out UBRRH,r16  
  
...  
; Set the USBS and the UCSZ1 bit to one, and  
; the remaining bits to zero.  
ldi r16,(1<<URSEL)|(1<<USBS)|(1<<UCSZ1)  
out UCSRC,r16 بیت برآورده نباشد
```

C Code Example

```
/* Set UBRRH to 2 */  
UBRRH = 0x02;  
  
/* Set the USBS and the UCSZ1 bit to one, and */  
/* the remaining bits to zero. */  
UCSRC = (1<<URSEL)|(1<<USBS)|(1<<UCSZ1);  
...
```



دسترسی به ثبات‌های UCSRC و UBRRH

دسترسی خواندن :

- خواندن ثبات‌های UCSRC و UBRRH با یک زمانبندی معین کنترل می‌شود
- خواندن یکبار مکان I/O محتوای ثبات UBRRH را برمی‌گرداند
- اگر مکان ثبات در سیکل ساعت قبلی خوانده شده باشد، خواندن ثبات در سیکل ساعت جاری، محتوای ثبات UCSRC را برخواهد گرداند
- دنباله زمانبندی برای خواندن UCSRC یک عملیات اتمیک می‌باشد
- عواملی چون وقفه نباید مانع از اجرای پشت سرهم و یک دفعه یک عملیات اتمی گردد.
- بنابراین وقفه‌ها در طی عمل خواندن باید کنترل شوند



نمونه کد به زبان اسembلی و C

Assembly Code Example

```
USART_ReadUCSRC:  
; Read UCSRC  
in r16,UBRRH  
in r16,UCSRC  
ret
```

C Code Example

```
unsigned char USART_ReadUCSRC( void )  
{  
    unsigned char ucsrc;  
    /* Read UCSRC */  
    ucsrc = UBRRH;  
    ucsrc = UCSRC;  
    return ucsrc;  
}
```



توصیف ثبات‌های USART

ثبات داده UDR-I/O USART

Bit	7	6	5	4	3	2	1	0	
Read/Write	R/W	UDR (Read)							
Initial Value	0	0	0	0	0	0	0	0	UDR (Write)

RXB[7:0]

TXB[7:0]

- ثبات بافر داده ارسال USART و دریافت USART، آدرس I/O یکسانی را به اشتراک می‌گذارند که به عنوان ثبات داده USART یا UDR شناخته می‌شود

- ثبات بافر داده ارسال TXB (USART) مقصدی برای داده نوشته شده در مکان رجیستر UDR است
- خواندن محل ثبات UDR، محتوای ثبات بافر داده دریافت (RXB) را برمی‌گرداند
- برای کاراکترهای ۵، ۶ یا ۷ بیتی، بیت‌های استفاده نشده بالاتر توسط ارسال‌کننده نادیده گرفته خواهند شد و توسط گیرنده صفر قرار داده می‌شوند



ثبت داده UDR-I/O USART

- بافر ارسال فقط وقتی پرچم UDRE در ثبات UCSRA یک است می‌تواند نوشته شود و زمانیکه یک نیست، توسط ارسال‌کننده USART نادیده گرفته خواهد شد
- زمانیکه داده در بافر ارسال، و ارسال‌کننده فعال است، ارسال‌کننده داده را در ثبات جابجایی ارسال بارگذاری می‌کند و داده به‌طور سریال روی پایه TXD منتقل می‌شود
- بافر دریافت شامل یک FIFO دو مرحله‌ای است. هر زمان که یک دسترسی به آن صورت گیرد، حالتی را تغییر می‌دهد. بنابراین از دستورات خواندن-ویرایش-نوشتن نباید استفاده کرد
- دستورات تست بیت (SBIS) و (SBIC) نیز حالت FIFO را تغییر می‌دهند

Bit	7	6	5	4	3	2	1	0	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	



کنترل USART و ثبات وضعیت UCSRA -A

بیت ۷ RXC، تکمیل دریافت USART :

- اگر داده خوانده نشده در بافر دریافت موجود بود، یک و اگر بافر خالی باشد صفر می‌شود
- اگر گیرنده غیرفعال باشد، بافر دریافت خالی و در نتیجه بیت RXC صفر خواهد شد
- پرچم RXC می‌تواند برای تولید وقفه تکمیل دریافت استفاده شود

بیت ۶ TXC، تکمیل ارسال :

- اگر تمام فریم در ثبات جابجایی به خارج انتقال یافته و هیچ داده جدیدی در بافر ارسال به طور جاری موجود نباشد، یک می‌شود
- بیت پرچم TXC زمانیکه یک وقفه تکمیل ارسال، اجرا می‌شود به طور خودکار صفر می‌گردد. می‌تواند با نوشتن یک در مکان بیت مربوطه صفر شود
- پرچم TXC می‌تواند یک وقفه تکمیل ارسال تولید کند



کنترل USART -A و ثبات وضعیت UCSRA

بیت ۵-UDRE، خالی بودن ثبات داده USART :

- نشان دهنده‌ی آمادگی بافر ارسال (UDR)، برای دریافت داده جدید است
- اگر UDRE یک باشد، بافر خالی است. پرچم UDRE می‌تواند یک وقفه خالی بودن ثبات داده تولید کند
- پرچم UDRE بعد از بازنشانی، برای نشان دادن آمادگی فرستنده USART، یک می‌شود

بیت ۴-FE، بیت خطای فریم :

- اگر پس از دریافت کاراکتر بعدی در بافر دریافت، خطای قاب داشته باشیم، یک می‌شود
- خطای قاب زمانیکه اولین بیت توقف از کاراکتر بعدی در بافر دریافت صفر باشد، رخداده و تا زمانیکه بافر دریافت (UDR) خوانده شود، معتبر است
- بیت FE وقتی بیت توقف یک باشد، صفر است. در زمان نوشتمن در UCSRA، این بیت را صفر کنید



کنترل USART-A و ثبات وضعیت UCSRA

بیت ۳- DOR، خطای سرروی به معنی خطای روی هم افتادگی داده‌های دریافتی :

- اگر یک وضعیت سرروی، تشخیص داده شود، این بیت یک می‌شود
- این بیت تا زمانیکه بافر دریافت (UDR) خوانده شود، معتبر است. زمان نوشتن در UCSRA، این بیت را صفر کنید

بیت ۲- PE، خطای توازن :

- اگر کاراکتر دریافتی یک خطای توازن داشته باشد و بیت چک کردن توازن (UPM=1) فعال باشد، این بیت یک می‌شود
- این بیت تا زمانیکه بافر دریافت (UDR) خوانده شود، معتبر است. این بیت را زمان نوشتن در UCSRA صفر کنید



کنترل USART -A و ثبات وضعیت UCSRA

بیت ۱- U2X، دو برابر کردن سرعت ارسال USART :

- فقط برای عملیات ناهمگام کاربرد دارد
- زمان استفاده از عملیات همگام، این بیت را صفر کنید
- یک کردن این بیت مقسوم علیه تقسیم کننده نرخ باد را از ۱۶ به ۸ کاهش خواهد داد

بیت ۰- MPCM، حالت ارتباط چند پردازشگری :

- حالت ارتباط چند پردازشگری را فعال می کند
- وقتی بیت MPCM یک می شود، همه فریم های ورودی دریافت شده که حاوی اطلاعات آدرس نیستند توسط گیرنده USART نادیده گرفته خواهند شد
- فرستنده از وضعیت بیت MPCM تاثیر نمی پذیرد



کنترل USART و ثبات وضعیت UCSRB

Bit	7	6	5	4	3	2	1	0	UCSRB
ReadWrite	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	

بیت ۷ RXCIE، فعالسازی وقفه تکمیل دریافت RX :

- یک کردن این بیت، وقفه ناشی از پرچم RXC را فعال و در صورتی که پرچم وقفه سراسری در SREG یک و بیت UCSRA در RXC یک باشد یک وقفه تکمیل دریافت USART تولید می شود.

بیت ۶ TXCIE، فعالساز وقفه تکمیل ارسال (TX) :

- یک کردن این بیت، وقفه ناشی از پرچم TXC را فعال و در صورتی که پرچم وقفه سراسری در SREG یک و بیت UCSRA در TXC یک باشد یک وقفه تکمیل ارسال USART تولید می شود



کنترل USART و ثبات وضعیت UCSRB -B

بیت ۵- UDRIE، بیت فعال‌ساز وقفه خالی بودن ثبات داده : USART

- یک کردن این بیت، وقفه روی پرچم UDRE را فعال و در صورتی که پرچم وقفه سراسری در SREG یک و بیت UDRE در UCSRA یک باشد، یک وقفه خالی بودن ثبات داده، تولید می‌شود

بیت ۴- RXEN، فعال‌ساز گیرنده :

- یک کردن این بیت، گیرنده USART را فعال می‌کند
- گیرنده وقتی فعال است کارکرد عادی پایه RXD به عنوان درگاه را متوقف می‌نماید
- غیرفعال کردن گیرنده، با فر دریافت را تخلیه و مقادیر پرچم‌های DOR، RE و PE را نامعتبر می‌سازد



کنترل USART و ثبات وضعیت UCSRB -B

بیت ۳، فعال‌ساز فرستنده : TXEN

- یک کردن این بیت، فرستنده USART را فعال می‌کند
- فرستنده وقتی فعال است کار کرد عادی پایه RXD به عنوان درگاه را متوقف می‌نماید
- غیرفعال کردن فرستنده، تا زمان تکمیل ارسال داده‌های در حال جریان و معطل کامل نشود، انجام نخواهد شد
- وقتی فرستنده غیر فعال شود، بعد از آن پایه TXD در وضعیت کار کرد عادی به عنوان بک بیت درگاه قرار می‌گیرد

بیت ۲، UCSZ2، طول کاراکتر :

- بیت‌های UCSZ2 در ترکیب با بیت‌های UCSZ1:0 موجود در UCSRA، تعداد بیت‌های داده را در یک فریم تنظیم می‌نمایند



کنترل USART و ثبات وضعیت UCSRB -B

بیت ۱- RXB8، بیت داده دریافتی شماره ۸ :

- نهمین بیت داده از کاراکتر دریافت شده در زمان انجام عملیات با فریم‌های سریال با ۹ بیت داده می‌باشد
- باید قبل از خواندن بیت‌های ۰ الی ۷، از UDR خوانده شود

بیت ۰- TXB8، بیت داده ارسالی شماره ۸ :

- نهمین بیت داده از کاراکتری است که باید در زمان انجام عملیات با فریم‌های سریال با ۹ بیت داده ارسال شود
- باید قبل از نوشتن بیت‌های بیت‌های ۰ الی ۷، در UDR نوشته شود



کنترل USART و ثبات وضعیت UCSRC

Bit	7	6	5	4	3	2	1	0	UCSRC
Read/Write	R/W								
Initial Value	1	0	0	0	0	1	1	0	

- ثبات UCSRC و ثبات UBRRH مکان I/O یکسانی را به اشتراک می‌گذارند

بیت ۷- URSEL، انتخاب ثبات :

- برای انتخاب بین دسترسی به یکی از ثبات‌های UCSRC یا UBRRH بکار می‌رود
- هنگام خواندن UCSRC این بیت یک خوانده می‌شود. زمان نوشتن در UCSRC این بیت باید یک باشد.

بیت ۶- UMSEL، انتخاب حالت USART :

UMSEL	Mode
0	Asynchronous Operation
1	Synchronous Operation



کنترل USART و ثبات وضعیت UCSRC -C

بیت ۴:۵ UPM1:0، حالت توازن :

- برای فعال کردن توازن و تنظیم نوع تولید و چک کردن توازن بکار می‌روند
- اگر فعال باشد، فرستنده، توازن بیت‌های داده در هر فریم را تولید کرده و می‌فرستد
- گیرنده یک مقدار توازن برای داده ورودی تولید و آن را با مقدار UPM0 مقایسه می‌کند. در صورت عدم تطابق، پرچم PE در UCSRA در پرچم PE در UCSRA یک خواهد شد

UPM1	UPM0	Parity Mode
0	0	Disabled
0	1	Reserved
1	0	Enabled, Even Parity
1	1	Enabled, Odd Parity



کنترل USART و ثبات وضعیت UCSRC -C

بیت ۳-**USBS**: انتخاب بیت توقف :

- تعداد بیتهاي توقف را که توسط فرستنده در انتهای يك فريم درج می شود، انتخاب می کند. گيرنده اين تنظيمات را ناديده می گيرد

USART Stop Bit Select	
USBS	Stop Bit(s)
0	1-bit
1	2-bit



کنترل USART و ثبات وضعیت UCSRC -C

بیت ۱: UCSZ1:0: طول کاراکتر :

- بیت‌های UCSZ1:0 در ترکیب با بیت UCSZ2 در ثبات UCSRB، تعداد بیت‌های داده را در یک فریم تنظیم می‌کنند

UCSZ2	UCSZ1	UCSZ0	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



کنترل USART و ثبات وضعیت UCSRC -C

بیت .UCPOL: قطبیت ساعت :

- تنها برای حالت همگام استفاده می‌شود. در حالت ناهمگام، این بیت را صفر کنید
- ارتباط بین تغییر خروجی داده و نمونه داده ورودی و ساعت همگام را تنظیم می‌کند

UCPOL	Transmitted Data Changed (Output of TxD Pin)	Received Data Sampled (Input on RxD Pin)
0	Rising XCK Edge	Falling XCK Edge
1	Falling XCK Edge	Rising XCK Edge



ثبتات نرخ باد USART شامل UBRRH و UBRRRL

اگر صفر لغایت شده UCSRC انتخاب رود

Bit	15	14	13	12	11	10	9	8	UBRRH	UBRRRL
	URSEL	-	-	-	UBRR[11:8]					
	UBRR[7:0]									
Read/Write	R/W	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		

- ثبات UBRRH مکان I/O یکسانی را به عنوان ثبات UCSRC به اشتراک می‌گذارند

بیت ۱۵: URSEL: انتخاب ثبات :

- برای انتخاب یکی از ثبات‌های UBRRH یا UCSRC بکار می‌رود
- زمان خواندن UBRRH، صفر خوانده می‌شود
- بیت URSEL در زمان نوشتمن در ثبات UBRRH، باید صفر باشد



ثبت نرخ باد USART و UBRRH شامل UBRRRL

بیت ۱۴:۱۲ - بیت‌های رزرو شده :

- بیت‌ها برای استفاده‌های آینده رزرو شده‌اند
- برای سازگاری با وسایل آینده، باید زمانیکه در UBRRH نوشته می‌شود، صفر باشند

بیت ۱۱:۰ - **UBRR11:0**: ثبات نرخ باد **USART** :

- نرخ باد USART را تعیین می‌کنند
- شامل چهار بیت با ارزش و UBRRRL شامل هشت بیت کم ارزش نرخ باد USART است

- در حال مخابره، تغییر نرخ باد، باعث آسیب داده‌های در حال انتقال می‌شود
- نوشتن در UBRRRL موجب به روز رسانی آنی پیش مقیاس‌گذار نرخ باد می‌شود



مثال‌هایی از تنظیمات نرخ باد

- خطای بالا بین نرخ باد واقعی و نرخ باد هدف قابل قبول است
- لیکن مقاومت گیرنده نسبت به نویز، در صورت وجود خطای بالا در نرخ باد، خصوصاً زمانی که طول فریم طولانی باشد، کمتر است
- مقدار خطا با استفاده از معادله زیر محاسبه می‌شود :

ریزپردازنده مبتنی بر منفذ باشد

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

- انتخاب حالت همگام و یا ناهمگام با توجه به نوع دستگاه جانبی صورت می‌پذیرد
- یکی از کاربردهای رایج واسط ارسال و دریافت سریال همگام-ناهمگام، دریافت اطلاعات محیط از طریق سنسورها و ارسال این اطلاعات به رایانه به منظور انجام پردازش‌های لازم بر روی این داده‌ها می‌باشد



مثال‌هایی از تنظیمات نرخ باد

Baud Rate (bps)	$f_{osc} = 1.0000 \text{ MHz}$				$f_{osc} = 1.8432 \text{ MHz}$				$f_{osc} = 2.0000 \text{ MHz}$			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	25	0.2%	51	0.2%	47	0.0%	95	0.0%	51	0.2%	103	0.2%
4800	12	0.2%	25	0.2%	23	0.0%	47	0.0%	25	0.2%	51	0.2%
9600	6	-7.0%	12	0.2%	11	0.0%	23	0.0%	12	0.2%	25	0.2%
14.4k	3	8.5%	8	-3.5%	7	0.0%	15	0.0%	8	-3.5%	16	2.1%
19.2k	2	8.5%	6	-7.0%	5	0.0%	11	0.0%	6	-7.0%	12	0.2%
28.8k	1	8.5%	3	8.5%	3	0.0%	7	0.0%	3	8.5%	8	-3.5%
38.4k	1	-18.6%	2	8.5%	2	0.0%	5	0.0%	2	8.5%	6	-7.0%
57.6k	0	8.5%	1	8.5%	1	0.0%	3	0.0%	1	8.5%	3	8.5%
76.8k	-	-	1	-18.6%	1	-25.0%	2	0.0%	1	-18.6%	2	8.5%
115.2k	-	-	0	8.5%	0	0.0%	1	0.0%	0	8.5%	1	8.5%
230.4k	-	-	-	-	-	-	0	0.0%	-	-	-	-
250k	-	-	-	-	-	-	-	-	-	-	0	0.0%
Max ⁽¹⁾	62.5 kbps		125 kbps		115.2 kbps		230.4 kbps		125 kbps		250 kbps	

1. UBRR = 0, Error = 0.0%



مثال‌هایی از تنظیمات نرخ باد

خطا معرفتی 😊

Baud Rate (bps)	$f_{osc} = 3.6864 \text{ MHz}$				$f_{osc} = 4.0000 \text{ MHz}$				$f_{osc} = 7.3728 \text{ MHz}$			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	95	0.0%	191	0.0%	103	0.2%	207	0.2%	191	0.0%	383	0.0%
4800	47	0.0%	95	0.0%	51	0.2%	103	0.2%	95	0.0%	191	0.0%
9600	23	0.0%	47	0.0%	25	0.2%	51	0.2%	47	0.0%	95	0.0%
14.4k	15	0.0%	31	0.0%	16	2.1%	34	-0.8%	31	0.0%	63	0.0%
19.2k	11	0.0%	23	0.0%	12	0.2%	25	0.2%	23	0.0%	47	0.0%
28.8k	7	0.0%	15	0.0%	8	-3.5%	16	2.1%	15	0.0%	31	0.0%
38.4k	5	0.0%	11	0.0%	6	-7.0%	12	0.2%	11	0.0%	23	0.0%
57.6k	3	0.0%	7	0.0%	3	8.5%	8	-3.5%	7	0.0%	15	0.0%
76.8k	2	0.0%	5	0.0%	2	8.5%	6	-7.0%	5	0.0%	11	0.0%
115.2k	1	0.0%	3	0.0%	1	8.5%	3	8.5%	3	0.0%	7	0.0%
230.4k	0	0.0%	1	0.0%	0	8.5%	1	8.5%	1	0.0%	3	0.0%
250k	0	-7.8%	1	-7.8%	0	0.0%	1	0.0%	1	-7.8%	3	-7.8%
0.5M	-	-	0	-7.8%	-	-	0	0.0%	0	-7.8%	1	-7.8%
1M	-	-	-	-	-	-	-	-	-	-	0	-7.8%
Max ⁽¹⁾	230.4 kbps		460.8 kbps		250 kbps		0.5 Mbps		460.8 kbps		921.6 kbps	

1. ۶۲ UBRR = 0, Error = 0.0%



مثال‌هایی از تنظیمات نرخ باد

Baud Rate (bps)	$f_{osc} = 8.0000 \text{ MHz}$				$f_{osc} = 11.0592 \text{ MHz}$				$f_{osc} = 14.7456 \text{ MHz}$			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	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. ^{۶۳} UBRR = 0, Error = 0.0%



مثال‌هایی از تنظیمات نرخ باد

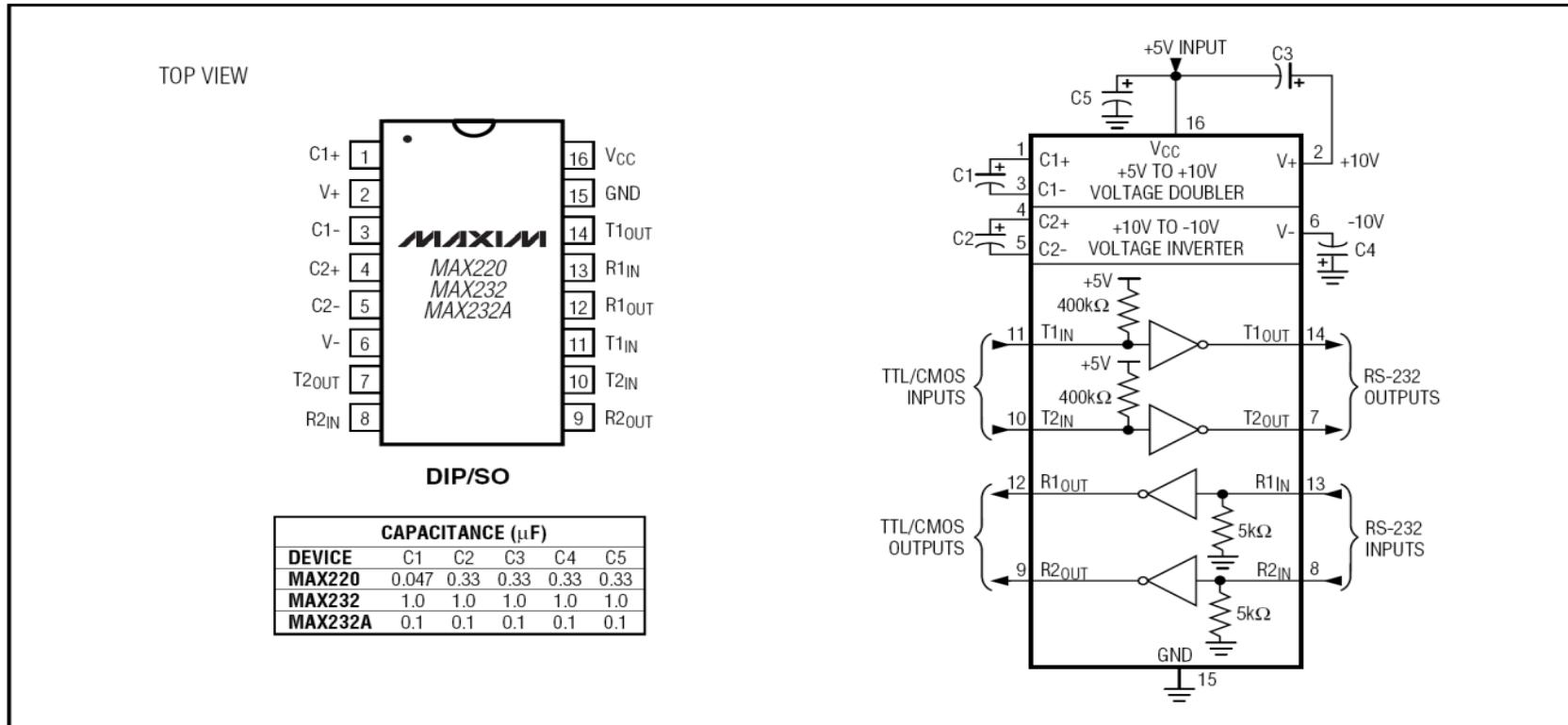
Baud Rate (bps)	$f_{osc} = 16.0000 \text{ MHz}$				$f_{osc} = 18.4320 \text{ MHz}$				$f_{osc} = 20.0000 \text{ MHz}$			
	U2X = 0		U2X = 1		U2X = 0		U2X = 1		U2X = 0		U2X = 1	
	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error	UBRR	Error
2400	416	-0.1%	832	0.0%	479	0.0%	959	0.0%	520	0.0%	1041	0.0%
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. ۶۴ UBRR = 0, Error = 0.0%



ارتباط RS232C از طریق پروتکل USART

- به منظور برقراری ارتباط سریال برای ارسال و دریافت همگام-ناهمگام اطلاعات بین تجهیزات مختلف، از تراشه‌هایی مانند قالب R232C استفاده می‌شود
- یکی از این تراشه‌ها تراشه **MAX232** می‌باشد :





توصیف ثبات‌ها

Bit	7	6	5	4	3	2	1	0	
Read/Write	R/W	UDR (Read)							
Initial Value	0	0	0	0	0	0	0	0	UDR (Write)

Bit	7	6	5	4	3	2	1	0	
Read/Write	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Initial Value	0	0	1	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
Read/Write	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8	UCSRB
Initial Value	0	0	0	0	0	0	0	0	

Bit	7	6	5	4	3	2	1	0	
Read/Write	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
Initial Value	1	0	0	0	0	1	1	0	



توصیف ثبات‌ها

Bit	15	14	13	12	11	10	9	8	UBRRH
	URSEL	-	-	-	UBRR[11:8]				UBRRL
					UBRR[7:0]				
Read/Write	R/W	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	